Stochastic schemes motivated by energetics and Riemannian geometry: optimization, Markov chain Monte Carlo, optimal control and data assimilation

Mariya Mamajiwala

A dissertation submitted in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

of

University College London.

Department of Statistical Science

University College London

February 8, 2023

I, Mariya Mamajiwala, confirm that the work presented in this thesis is my own. Where information has been derived from other sources, I confirm that this has been indicated in the work.

Abstract

Statistical algorithms not only involve drawing realizations from a given distribution or estimating the parameters of the related density, but a wider class of problems such as optimal control, data assimilation and non-convex optimization. Unlike a deterministic search algorithm, e.g. one based on quasi-Newton updates, stochastic search schemes can make use of concepts from both deterministic dynamics and stochastic theory of noise to strike a balance between exploitation and exploration in the design space. In the quest for more efficacious schemes, researchers have drawn on ideas from contemporary physics and differential geometry in arriving at suitably constrained dynamical systems that guide the search, and the work in this dissertation is similarly inspired. To start with, a survey of the state-of-the-art is presented in Chapter 1 to motivate and put in perspective the work in the chapters to follow. Chapter 2 dwells on a Riemannian geometric approach to non-convex optimization, wherein the flow that minimizes a given objective function with progressing iterations is constrained to live on a manifold defined using a metric derivable by treating the objective function as energy. Specifically, the underlying dynamical system is designed as a geometrically adapted Langevin stochastic differential equation (SDE). The same adaptation, albeit with a Riemannian metric given by the Fisher information matrix obtainable from the available likelihood, is used in Chapter 3 to arrive at an MCMC method. In Chapter 4, a time-recursive scheme for stochastic optimal control is proposed using SDEs integrated strictly forward in time, thus bypassing the computationally inexpedient forward-backward route to solve the Hamilton-Jacobi-Bellman (HJB) equation. We address the combined state-parameter estimation problem via stochastic filtering in Chapter 5, with a new

proposal for the parameter dynamics for higher accuracy and faster convergence. The thesis is concluded in Chapter 6 with a summary and scope for future research.

Impact Statement

The most significant contribution of this thesis is perhaps in laying out a novel stochastic approach to pose and solve a broad class of non-convex optimization problems within a Riemannian geometric setting. Specifically, it is shown that a Riemannian structure to the optimal search may be brought forth via the objective function alone and the stochastic update is implemented using time integration of an appropriately constructed Langevin dynamics developed on the Riemannian manifold. While such a geometric makeover significantly accelerates the search process, the stochastic update avoids the pitfall of solutions being stuck around local extrema. Given the ubiquity of non-convex optimization across a broad swathe of applications of industrial interest and the ready adaptability of the method to optimization with constraints, the novel perspective offered in our proposal should constitute a noteworthy scientific advance in this much researched area. It might also be usefully exploited in neural network schemes where optimization is extensively used to determine the learning parameters. By way of a specific application of a fairly general appeal, we have used the new optimization paradigm to propose a Riemannian geometric makeover of the MCMC method, widely acclaimed as one of the top ten algorithms of the 20th century. We have implemented the new scheme for identifying the unknown parameters of a posterior probability density from a given set of realizations – a problem of central interest in statistical model calibration, among others. Through a carefully chosen set of numerical examples, the salient advantages of a properly constructed manifold-based search are once more brought forth vis-à-vis the classical route and a few other existing schemes that have also aimed at organizing manifold-based searches. This thesis has also described a novel and computationally efficacious strategy for stochastic optimal control (SOC) of dynamical systems including mechanical oscillators, where the control term is linear. Such problems routinely appear in myriad practical applications such as

option pricing, risk hedging, insurance and robotics. The proposal is based on a predictive approximation to the terminal cost using Ito's formula, thereby obviating the need for costly and repetitive evaluations of forward and backward solutions at every update. Whilst ensuring a minimal overall cost, our posing of the SOC problem may however yield a different path vis-à-vis the conventional setting. Finally, we have considered the problem of estimation of states and parameters of dynamical systems within the framework of stochastic filtering. With certain energy-based arguments, our contribution has been in tweaking the prediction equations towards a faster and more accurate search for the parameters. Given the extensive use of stochastic filtering algorithms for real-time data assimilation in atmospheric modelling and other areas, the proposed parameter dynamics should have its share of usefulness in system identification problems of great practical import.

The Riemannian geometric and energy-based principles emphasized in this work for developing statistical algorithms are perhaps a step forward in a more nuanced and rational exploitation of advanced physical theories, in line with the ongoing research focus worldwide in the same direction.

Acknowledgements

There are many people without whose active roles, completion of this thesis might not have come about. Most importantly, I wish to thank my supervisor, Professor Serge Guillas - for reposing unshakeable faith in me and helping me secure a scholarship, for his supervisory guidance enhanced with a personal touch, for his ever so thorough feedback, and for providing me with more support and resources than I think I deserved. His Pollyannish outlook has always left me motivated, no matter how dim the situation (and of course there have been more than a few of those!).

It would be an injustice if I did not thank Professor Debasish Roy, Indian Institute of Science - first for giving me the opportunity to recover from a career break, for introducing me to Professor Serge Guillas, and most importantly, for the many hours he spent teaching me the fundamentals of differential geometry without which this thesis might have taken a very different course.

Next, I wish to thank my co-supervisors Dr. Robert Wicks, and Dr. Alexandros Beskos for playing the supporting role that I could always count on; Dr. Samuel Livingstone, for spending his valuable time and providing useful feedback on my interim report; Dr. G V Rao, with whom help was always just a phone-call away, and whose zeal for work at the age of more than 75 I find truly inspiring.

Thanks are in order for the following funding bodies that supported me throughout my time as a PhD student: EPSRC for the full 4-year studentship; the Alan Turing Institute for hosting me as a visiting researcher; the Centre of Excellence in Advanced Mechanics of Materials (part of the Indian Institute of Science, Bangalore) for funding my stay at their campus for several months over the past four years; the Institute of Mathematical Statistics, the International Society for Bayesian

Analysis, and the Mathematisches Forschungsinstitut Oberwolfach for their travel grants that enabled me to participate in various conferences. I am ever so grateful to Professors Ilya Pavlyukevich and Vladimir Matveev of the Friedrich Schiller University, Jena, Germany for welcoming me as a visiting researcher, taking the time for some wonderful discussions and also bearing the cost of my access to the university facilities.

I must thank the late Ms Deepti Jayawardena-Wilkinson who helped me find my bearings when I started out. Thanks are also due to Dr. Paul Northrop and Dr. Terry Soo who were there to instil confidence in me when things seemed uncertain; to all the staff for bearing with me whilst I continued to work remotely even as most others were back in office. I am grateful to all the professors with whom I have worked as a teaching assistant, which melded work with the opportunity to learn the art of teaching. I am thankful to my labmates for the many discussions that we have had - Dr. Devaraj Gopinathan in particular, the most dependable friend and colleague one could hope to have.

I am indebted to my mother for the innumerable sacrifices she made for me and for bringing out the winner in me, and my father who had already taught me everything I needed to become a good learner during my school days, much more than I have come to learn after reading a multitude of self-help books. Finally, I cannot even find words to thank my husband who has been my best friend and at times my mentor, who believed in me when I wavered, supported me every step of the way - to him, I owe everything.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

A few of the most successful methods in statistical analysis are inspired from physics; a couple of examples being the simulated annealing method of optimization inspired by thermodynamics and the Hamiltonian Monte Carlo (HMC) method of Markov chain Monte Carlo (MCMC) based on Hamiltonian mechanics. That simulated annealing and HMC are known to be efficient owes largely to the fact that the underlying concepts are drawn from classical particle physics. However, after a review of the state-of-the-art in many of these methods, an important observation may be made. There remains considerable scope to exploit concepts from Riemannian or non-Euclidean geometry of manifolds to develop new methods that are more informed and hence more efficient, fast and accurate.

Bearing this in mind, and in an attempt to exploit the power of physics, stochastic analysis and differential geometry simultaneously, this thesis proposes novel methods in optimization, MCMC, stochastic optimal control (SOC) and parameter estimation via data assimilation (DA). Towards this end, one of the fundamental principles used is stochastic development - a concept for the treatment of stochastic differential equations (SDEs) on Riemannian manifolds (RM). Another theme that can be seen recurring in this thesis is the use of some aspects of energetics to define a pseudo-energy surface on which the solution of the problem is sought. Yet another feature common across all the methods discussed is that of Monte Carlo simulations, i.e. using a finite number of samples to approximate the desired probability distribution.

Indeed, many problems in statistical analysis are or may be construed as those of either optimization or probabilistic sampling. This is true in particular for the methods that are discussed in this thesis, viz. finding the global extremum of a non-convex function, MCMC, SOC and combined state and parameter estimation via data assimilation. Interestingly enough, it has been shown that the problems in optimization and sampling are related [1]. It basically has to do with the fact that the functions $f(x)$ and $\frac{f(x)}{\tau}$ have the same minima, and the Langevin Monte Carlo algorithm for $\frac{f(x)}{\tau}$ in the limit $\tau \to 0$ reduces to the gradient descent method for $f(x)$. Moreover, the problems of optimal control and data assimilation also have an interesting relationship as discussed in this introduction. In what follows, the problem statement and the proposed methodology for the next four chapters is given in brief. Moreover, the state-of-the-art is also described in a simple language for each of these methods; a detailed literature survey and the technical details for these methods are provided in the respective chapters. The thesis is concluded in Chapter 6 with a summary and a few possible directions for future research.

Chapter 2 discusses a method of unconstrained non-convex optimization. We focus on continuous optimization problems. The method basically consists of integrating an SDE, in particular the annealed Langevin diffusion equation [2], on an RM. First, this chapter provides the derivation of a general SDE on an RM (based on the concept of stochastic development) on which the optimization scheme directly depends. The concept of stochastic development is the one that is central to the construction of the Brownian motion (which is an SDE with a trivial drift) on an RM. We extend this approach to an SDE with a non-zero drift, thus obtaining an equation of a general SDE on an RM. There are two basic differences between the proposed method and the methods of optimization on an RM available in the literature. The first is that while the existing methods are deterministic, the proposed one is stochastic. The second is that, existing RM methods are only meant for constrained optimization problems wherein the constraint surface is itself treated as an RM. We, however, propose a way of defining an RM for an unconstrained problem which is discussed in the chapter. Now, evolutionary optimization schemes like

genetic algorithms [3] consist of some of the most successful methods of global optimization. One aspect worth appreciating - when compared to the evolutionary schemes - is that while the evolutionary schemes have an additional randomization step to facilitate exploration of the search space, this randomness occurs naturally in the proposed method owing to the inherent stochasticity of the equation being used. Another distinction between the proposed and evolutionary schemes is that while the former depends on gradient information of the objective function, most evolutionary schemes are gradient-free. It thus has directional information to facilitate better exploitation of the search space.

In Chapter 3, we discuss the application of the SDE on an RM as derived in Chapter 2 to MCMC. An MCMC problem consists of drawing samples from a desired density. These may then be used to approximate complicated integrals (e.g. determine the expectation of certain functions of a random variable) or other quantities when analytical solutions to such problems are unavailable. The methodology as proposed may be considered as the RM counterpart of the Metropolis adjusted Langevin Algorithm (MALA) [4]. Algorithm-wise, it is similar to that of the optimization algorithm in Chapter 2, except for a couple of points. The first is, the metric is now given by the Fisher-Information matrix (FIM) for the class of problems considered (statistical estimation of parameters). The second is the absence of annealing and hence the presence of an accept-reject step in line with the Metropolis-Hastings algorithm. Indeed, it is possible to get away with the second difference, in which case the algorithm may be considered as the RM counterpart of the unadjusted Langevin algorithm (ULA) [2]. However, we chose to work with the accept-reject step as it is easier to compare the proposal with several other methods of MCMC on RM from the literature. The aim of this chapter, in addition to providing a novel MCMC method, is to shine light on some of the issues with the existing methods. These are discussed in Section 3.4. The observations thus made are reinforced with some examples involving a few basic probability distributions, demonstrating that our method is the only one able to converge in moderate to high dimension. This chapter also affords insights into the Fisher-Information being an

appropriate metric for problems in statistical parameter estimation.

Chapters 4 and 5 are about SOC and, parameter estimation via DA respectively; these problems have an interesting relationship, i.e. they may be looked upon as two sides of the same coin. For an optimal control problem, the objective is to determine an additional control force to the noisy process dynamics in order to realize certain objectives. e.g. to reach a desired terminal state whilst minimizing a certain total cost. On the other hand, the objective of DA (of states alone or states and parameters; see Sections 5.2 and 5.3) is also to find an additional force-like term in the noisy process dynamics; the aim now is to determine the true evolution of certain states (that have already evolved in time) by making use of the available measurements that may be partial (i.e. not all states may be measured) and noisy (owing to instrumentation error). The problem of DA (and, by analogy, also SOC) is one of sampling from the posterior distribution. The data are the measurements available, while the dynamics of measurements defines the likelihood. The prior is defined by the dynamics of the system under consideration. Hence, this is essentially a Bayesian problem whose solution is the evolution of the posterior probability. The fundamental difference between this framework and MCMC then, is that, in MCMC the posterior remains frozen in time, whereas it does in the case of data assimilation thus making it more difficult to solve.

The method of SOC in Chapter 4 is derived for a class of problems referred to in the community as linear-quadratic (LQ) control, which is defined in Section 4.3. This is the most commonly encountered class of problems, and this choice is crucial to the derivation of the proposed method. It is well-known that the solution of an SOC problem may be associated with that of a partial differential equation (PDE) with a terminal condition; particularly the Hamilton-Jacobi-Bellman equation (HJB) [5]. When the HJB equation is semilinear and satisfies certain other constraints (discussed in Chapter 4), it can be solved via a nonlinear version of the Feynman-Kac formula [6]. According to this approach, the solution to the PDE may be obtained by simulating an associated pair of partly coupled forward-backward stochastic differential equations (FBSDEs). This method can be exploited for a

certain class of SOC problems wherein the associated HJB equation is semilinear and satisfies certain constraints. Although an elegant way to interpret and solve a PDE, simulating the system of FBSDEs can be computationally demanding. In our work, the HJB equation pertaining to the SOC problem is reformulated such that instead of the given terminal condition, the proposed PDE is now subject to a corresponding initial condition. Associated with the PDE, we then derive a set of SDEs whose solutions move only forward in time. This approach has a significant computational advantage over the original formulation. Moreover, since the method of FBSDEs generally requires simulating SDEs from the current to the terminal time at every step, the integration errors may accumulate and carry forward in estimating the control force. This error is particularly high initially, since the time of integration is the longest there. The proposed method bypasses such errors and is thus more robust. Unsurprisingly, it also exhibits a lower sampling variance. In addition to the computational superiority, this approach has another advantages over the traditional FBSDEs. First, the control may be started at any intermediate time - this being particularly beneficial for chaotic systems as explained in the chapter - which is not possible for FBSDEs. Second, the terminal time for the proposed method may be flexible, which is again not possible for the FBSDEs. This is because the HJB equation is subjected to a terminal condition, thus making the knowledge of terminal time imperative.

The problem of DA or stochastic filtering is to estimate the posterior distribution of states of a dynamical system given a noisy physics model and noisy measurements. Chapter 5 is about DA in general. In particular, however, our focus is on the estimation of parameters of a dynamical system within the data assimilation framework. Typically, while solving this type of problem, the unknown parameters of the system are appended to the state vector and data assimilation is performed over this combined state and parameter vector. The parameters do not really have any physical evolution, hence their evolution is taken simply as Brownian motion - this forms the noisy physics model for the parameters. It is worth noting that stochasticity is essential in the evolution of states as well as parameters without which the

machinery of data assimilation may break down. Now, the Brownian motion proposal for parameter dynamics, being quite random, has certain disadvantages. In Chapter 5, we construct a proposal for parameters to replace Brownian motion. We draw inspiration from Chapter 2 and define a pseudo-energy term and construct a Langevin diffusion equation using this energy. We provide one possibility for the definition of pseudo-energy using measurements, however, it is quite flexible as discussed further in the chapter. The motivation behind using an energy to define the Langevin diffusion (in Chapter 2 as well as Chapter 5) comes from the canonical ensemble in statistical mechanics [7]. Specifically, consider a mechanical system immersed inside a heat bath at a given temperature; then the canonical ensemble is the statistical ensemble of all possible states of the mechanical system when it is in thermal equilibrium with the heat bath. Each distinct microstate of the canonical ensemble is assigned a probability P which is related to the total energy (E) of that microstate as follows:$P \propto \exp(-E)$. Thus, the $\nabla \log P$ in the Langevin diffusion may be considered as $-\nabla E$, thereby allowing one to construct Langevin diffusion using energy instead of probability. We discuss the proposed approach in detail and illustrate its performance on a few examples.

Chapter 2

Stochastically developed Langevin dynamics: derivation and application to non-convex optimization

This chapter is largely based on the published work in [8] with emphasis on the application of stochastic development to non-convex optimization.

2.1 Introduction

At their core, most problems in statistical analysis are those of optimization. These problems may either be convex or non-convex. Convex optimization problems have a unique extremum and are hence simpler to solve. The majority of continous optimization problems however, fall in the category of non-convex functions; examples range from matrix completion problems[9] to neural networks[10]. Non-convex problems are more difficult to solve for a number of reasons such as the existence of multiple extremas, saddle points, presence of very flat regions or regions of widely varying curvature. In this chapter, our interest is in continuous non-convex problems. There are well-established methods for solving convex problems such as the gradient descent [11], conjugate gradient [12], Newton and quasi-Newton methods like the Davidon-Fletcher-Powell (DFP) and Broyden-Fletcher-Goldfarb-Shanno (BFGS) [13]. The success of methods for non-convex problems on the other hand relies on a good exploration-exploitation trade-off. Loosely put, ex-

ploitation aims at quickly finding a local minimum given an initial point, whereas exploration is aimed at searching the space for a different initial point which will potentially be closer to a better local minima. [14] is a gradient-based method which is well-known for global optimization. It generalizes Nesterov's accelerated gradient method [15] for convex optimization problems to non-convex problems. A majority of methods for global optimization, however, fall in the broad class of gradient-free evolutionary schemes that are based on biological heuristics and may not necessarily have a mathematical basis. Examples of these include, but are not limited to, genetic algorithms [3], particle swarm optimization [16], ant colony optimization [17], differential evolution [18] , covariance matrix adaptation [19, 20]. An interesting exception to these derivative-free methods is the replica exchange method [21] wherein a deterministic gradient descent is aided by stochastics to facilitate exploration. Specifically, the deterministic gradient descent and Langevin diffusion equation are evolved simultaneously and as soon as the Langevin trajectory gives a better solution, the starting points for both these dynamics are swapped. We propose a method that is also based on the Langevin diffusion, however, it is devised by marrying the concepts from stochastics and differential geometry alongwith a bit of physics. This is aimed at utilizing the best of the two subjects in a seamless manner thus bypassing the need for an additional step or intervention to help with exploration of the solution space.

While there are Riemannian manifold (RM) versions of several well-known methods for non-convex optimization such as the steepest descent, conjugate gradient and Newton's methods [22], these are all deterministic methods applicable only to constrained problems and posed in a way where the constraint surface is itself the manifold. We propose, perhaps for the first time, a method based on Riemannian geometry for optimization problems without any constraints. This may also be the first time that a stochastic method of non-convex optimzation is proposed using Riemannian differential geometry.

Towards this, we first derive a stochastic differential equation (SDE) on an RM using the concept of stochastic development as per [23]. Next, we make use of ener-

getics to arrive at a possible Riemannian metric. The Whitney embedding theorem [24], guarantees an embedding of any RM within a sufficiently higher dimensional Euclidean space. Characterizing the embedding space is however no trivial task in general. Since embedding within a higher dimensional Euclidean manifold is generally infeasible, the understanding of diffusion on a manifold that is intrinsically defined must be through the use of frame bundles [25]. As an alternative and with inspiration drawn from the work in [23], we emply the framework of stochastic development that is used for the derivation of the equation for Brownian motion on an RM, leading to the celebrated Laplace-Beltrami operator. We essentially extend this approach for a general SDE, i.e. an SDE with a non-trivial drift. We then apply this result to determine the equation of Langevin diffusion on an RM which is used for optimization in this chapter and Markov chain Monte Carlo in the next. This result is also applied to other interesting problems like developing a stochastic Hamiltonian preserving numerical integration scheme as in [8]. We limit our work to unconstrained problems for now, but it may not be too difficult to recast any constrained problem as an unconstrained one by using for instance the method of Lagrange multipliers (augmented Lagrangian) or penalty functions. Therefore, the proposed methods may also be extended to be used for problems with constraints.

The rest of this chapter is organized as follows. Section 2.2 describes the concept of stochastic development after a brief review of the relevant background. In Section 2.3, we provide the derivation for a special case via a different route to arrive at the same result. Finally, we illustrate the proposed method on a few benchmark problems in non-convex optimization in Section 2.4 and compare the results thus obtained with a few other competing algorithms before concluding the work in Section 2.5.

2.2 Stochastic development on a Riemannian Manifold

By way of a ready reference, we give brief reviews of concepts in stochastic calculus and differential geometry in sections 2.2.1 and 2.2.2 respectively. The main result is

provided in Section 2.2.3, where we use the notion of frame bundles on Riemannian manifolds to stochastically develop an SDE with a non-trivial drift.

2.2.1 A brief review of stochastic calculus

Stochastic calculus deals with the properties of solutions of SDEs. A typical form of an SDE is as follows:

$$dx_t = \alpha(x_t,t)dt + \beta(x_t,t)dB_t \tag{2.1}$$

Here, x_t represents the stochastic process, $\alpha(x_t,t)$ and $\beta(x_t,t)$ are vector and matrix valued functions, respectively and dB_t is the Brownian vector increment at time t, with all its scalar components being independent of one another. In the above equation, $\alpha(x_t,t)dt$ is referred to as the drift term and $\beta(x_t,t)dB_t$ as the diffusion term. If we remove the diffusion term from (2.1), it reduces to an ordinary differential equation (ODE). It is worth emphasizing that the Brownian motion B_t is everywhere continuous but nowhere differentiable and this calls for an approach different from the standard calculus in $\mathbb{R}^N$ in solving an SDE. The pursuit of stochastic calculus essentially begins with the interpretation of the diffusion (stochastic) integral $\int_{t_0}^{t}\beta(x_s,s)dB_s$. There are mainly two versions of stochastic calculus, viz. Ito and Stratonovich, and it is readily possible to switch between the two. In Ito's calculus, the stochastic integral is interpreted as

$$\int_{t_0}^{t}\beta(x_s,s)dB_s = \lim_{max_i(t_{i+1}-t_i)\to 0}\sum_i \beta(x_{t_i},t_i)(B_{t_{i+1}}-B_{t_i}) \tag{2.2}$$

where $t_0 < t_1 < ... < t_i < ...$ is a discretization of the closed interval $[t_0,t]$. In Stratonovich calculus, on the other hand, this integral is interpreted as

$$\int_{t_0}^{t}\beta(x_s,s)dB_s = \lim_{max_i(t_{i+1}-t_i)\to 0}\sum_i \frac{1}{2}(\beta(x_{t_i},t_i)+\beta(x_{t_{i+1}},t_{i+1}))(B_{t_{i+1}}-B_{t_i}) \tag{2.3}$$

While the Ito version has the physical appeal of causality built into its construction, the Stratonovich version conforms better with the features of standard calculus in $\mathbb{R}^N$. We interpret the solution of SDEs in this work in Ito's sense. The basic ingredi-

ent of this calculus is Ito's formula which we now describe. Consider the stochastic process $x(t)$ which is the solution of the following SDE:

$$dx(t) = \mu(t)dt + \sigma(t)dB(t) \tag{2.4}$$

If $f(x)$ is a twice continuously differentiable function of x, then Ito's formula gives the following SDE for $f(x(t))$:

$$\begin{aligned} df(x(t)) &= f'(x(t))dx(t) + \frac{1}{2}f''(x(t))d[x,x](t) \\ &= (f'(x(t))\mu(t) + \frac{1}{2}f''(x(t))\sigma^2(t))dt + f'(x(t))\sigma(t)dB(t) \end{aligned} \tag{2.5}$$

In the equation above, $[x,x](t)$ denotes the quadratic variation of $x(t)$ and is defined as

$$[x,x](t) = \lim_{\delta_r \to 0} \sum_{i=1}^{r} ||(x(t_i^r) - x(t_{i-1}^r))||^2 \tag{2.6}$$

where this limit is taken over the set of all possible partitions:

$$0 = t_0^r < t_1^r < t_0^r < ... < t_r^r = t \quad \text{with} \quad \delta_r = \max_{1 \le i \le r} (t_i^r - t_{i-1}^r)$$

One of the most remarkable results in the theory of stochastic calculus is that the quadratic variation of the Brownian motion is $[B,B](t) = t$ with probability 1; the result is remarkable since, although Brownian motion is stochastic, its quadratic variation returns a strictly deterministic quantity [6]. Mainly owing to non-linearity in the drift and/or diffusion terms, an analytical solution to an SDE is generally not available. Solutions in general must therefore be obtained through various numerical integration schemes, such as the Euler-Maruyama [26].

2.2.2 A brief review of concepts from differential geometry

Differential geometry is the mathematical machinery for performing calculus over an arbitrarily shaped hypersurface in any dimension, say $\mathbb{R}^d$ and can be seen as a useful generalization of standard calculus in the Euclidean setting. The departure from the Euclidean set-up is specifically captured through certain incompatibility

tensors, e.g. the curvature tensor in Riemannian geometry. A small neighbourhood around every point in the hypersurface, which is referred to as the manifold, is represented by a local co-ordinate chart, possibly drawn from the embedding Euclidean space. The embedding Euclidean space is of a strictly higher dimension, say $\mathbb{R}^n$ with $n > d$. These local charts overlap smoothly to enable calculations on the manifold as a whole. An important concept in the theory of differential geometry is that of a tangent plane. As the name suggests, it is the unique plane tangent to the manifold at a given point. Formally, a manifold is called Riemannian if the tangent plane at every point p is equipped with an inner product with respect to a given metric g such that, if X_p and Y_p are two vectors on the tangent plane, we have

$$\langle X_p, Y_p \rangle = [g_p]_{ij} x^i y^j \tag{2.7}$$

where $X_p = x^i e_i$, $Y_p = y^j e_j$; $\{e_i\}_{i=1}^d$ being the canonical basis vectors in $\mathbb{R}^d$.

In the Euclidean setting, we have $g_{ij} = \delta_{ij}$ where δ_{ij} represent the Kronecker delta symbols. Loosely speaking, g encapsulates the notion of how distances and angles between two vectors are measured on a tangent plane. It is known that every RM is associated with a unique Riemannian metric. Now that we have seen that every point on the RM has a tangent plane attached to it and that every tangent plane in turn has a unique metric, one must also figure out a way to smoothly move from one tangent plane to another in a close neighbourhood of the former (parallel transport of vector and tensor fields). This is precisely where the concept of a connection comes in. For a given Riemannian metric g, the coordinate representation of the connection is given as

$$\gamma_{ij}^k = \frac{1}{2} g^{kl} [\partial_i g_{jl} + \partial_j g_{il} - \partial_l g_{ij}] \tag{2.8}$$

In the above equation, $g^{kl} = g_{kl}^{-1}$ and the symbols γ_{ij}^k are also referred to as the Christoffel symbols. It must be noted that γ is not a tensor, as it does not transform like one under a smooth change of co-ordinates. The usual concept of derivatives of vectors in $\mathbb{R}^n$ does not apply to the RM, since any two vectors lying in two different

tangent planes are objects of different vector spaces, and hence cannot be added or subtracted in the usual way. The equivalent notion of derivative on the RM is known as covariant derivative and it is defined in terms of the connection. The covariant derivative of a vector Y along a vector X in terms of the Christoffel symbols is defined as follows:

$$\nabla_X Y = [XY^k + X^i Y^j \gamma_{ij}^k] e_k \tag{2.9}$$

where $X = X^i e_i$, $Y = Y^j e_j$, e_i is the unit vector in the i^{th} co-ordinate direction in terms of a local chart. We emphasize that equation (2.9) is valid only within the cut locus; roughly speaking the cut locus at a point p on the manifold is that neighbourhood (on the manifold), every point in which has a geodesic connecting the point p (see below for the definition of a geodesic on the RM).

Now that we have a way of moving from one point on the manifold to another using the connection, we can define curves. An important example of a curve on the manifold, parametrized by t, is that of a geodesic. It is the shortest path joining two given points on the manifold. The equation of a geodesic is as follows:

$$\ddot{x}^k(t) + \dot{x}_t^i \dot{x}_t^j \gamma_{ij}^k(x(t)) = 0 \tag{2.10}$$

The Euclidean equivalent of the above equation is just $\ddot{x}^k(t) = 0$, solutions to which are straight lines.

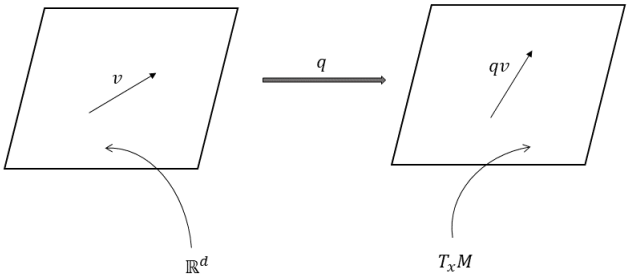
2.2.3 The concept of stochastic development

We may combine the basics of stochastic calculus with differential geometry to recast an SDE, originally posed in a d dimensional Euclidean space, on an RM M of the same dimension. A systematic framework for this is provided by stochastic development, which has been used in [23] to recast a Brownian motion on M. We presently use a similar strategy for SDEs that have a non-zero drift. In order to relate the canonical d dimensional Euclidean basis to a basis of the tangent plane $T_x M$ at the point $x \in M$, we need an additional construct of a $d + d^2$ dimensional manifold called the frame bundle $F(M)$. While the d-dimensional component of

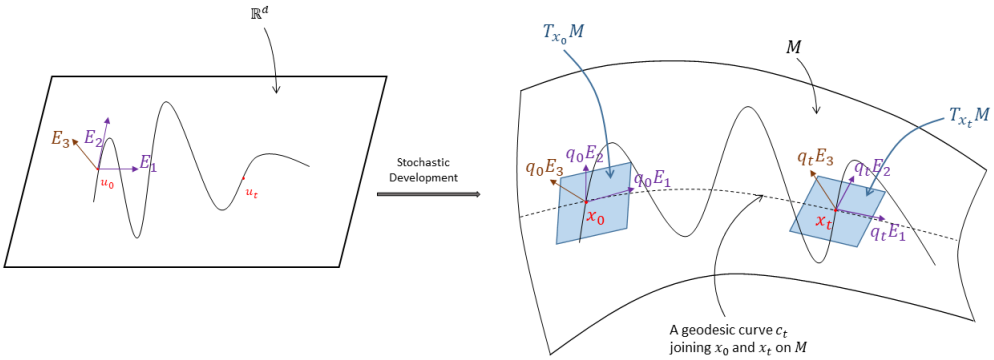
$F(M)$ is the base manifold M itself, the remaining d^2-dimensional part corresponds to orthogonal linear transformations applied to vectors on T_xM. We now reflect on how the connection ∇ on M manifests itself on the frame bundle $F(M)$. Clearly, a frame at a point $x \in M$ provides a linear isomorphism between the Euclidean space $\mathbb{R}^d$ where the solution of a standard SDE evolves and the d-dimensional tangent plane T_xM to M on which the solution needs to be projected. Thus, it is through the frame bundle that we can track these paths on M once we know how it evolves in $\mathbb{R}^d$. Let $E_1, ..., E_d$ be the co-ordinate basis vectors of the d-dimensional Euclidean space. Now, considering a frame q at x, we note that the vectors $qE_1, ..., qE_d$ make up a basis for T_xM.

We denote by $F(M)_x$ the set of all frames at x so that the elements of $F(M)_x$ may be acted upon by $GL(d,\mathbb{R})$, the general linear group. This means that any linear transformation of $F(M)_x$ is also a valid frame at x. $F(M)_x$ is also called a fibre at x. However, the base manifold M is presently Riemannian so that the torsion tensor is zero, and thus an orthonormal frame remains orthonormal upon parallel transport along M. There is thus no loss of generality in restricting the general linear group to the orthogonal group $O(M)$. Roughly speaking, a fibre $\mathscr{F}_x$ at a point x on M is defined as a space attached to that point. We may now define a surjective or onto map $\pi : \mathscr{F}(\mathscr{M})_x \longrightarrow M$. We define the frame bundle as the union of sets of frames at different points on the manifold, i.e. $F(M) = \bigcup_{x\in M} F(M)_x$. At this stage, we may actually look upon $F(M)$ itself as a (differentiable) manifold of dimension $d + d^2$. Accordingly, the projection map $\pi : F(M) \longrightarrow M$ is also smooth. Now we consider a point $q \in F(M)$ and the associated tangent space $T_qF(M)$ at the same point. It is a vector space of dimension $d + d^2$. We refer to a tangent vector $Y \in T_qF(M)$ as vertical if Y is tangent to the frame $F(M)_{\pi q}$. These vertical tangent vectors form a subspace $V_qF(M)$ of $T_qF(M)$ and it is of dimension d^2. Let the base manifold M be equipped with a Riemannian connection ∇. Then a curve q_t in $F(M)$, which is basically a smoothly varying field of frames, could be projected to a smooth curve $x_t = \pi q_t$ on M. We call the frame field q_t horizontal if the vector field q_tE is parallel along the projected curve x_t on the base manifold M for an arbitrary vector $E \in \mathbb{R}^d$.

We recall here that a vector field V along the curve x_t on M is called parallel along x_t if $\nabla_{\dot{x}} V = 0$ for every t. This is just an extension of the notion of parallel vectors in the Euclidean setting. The vector V_{x_t} at x_t is the parallel transport of the vector V_{x_0} at x_0.



(a) q is an isomorphism between $\mathbb{R}^d$ and T_xM - the tangent space at x on M



(b) The horizontal motion of q_t on M : The tangent vectors $(q_0E_1,\ q_0E_2)$ at $T_{x_0}M$ are parallelly transported according to the curvature of M

Figure 2.1: Stochastic development : a schematic illustration

We call a tangent vector $X \in T_qF(M)$ horizontal if it is tangent to the horizontal curve q_t. The space of horizontal vectors at q is denoted by $H_qF(M)$; it is a subspace

of $T_qF(M)$ and is dimension d. We thus have the direct-sum decomposition

$$T_qF(M) = V_qF(M) \oplus H_qF(M)$$

. Using the projection $\pi : F(M) \longrightarrow M$, a pushforward operation $\pi_* : H_qF(M) \longrightarrow T_xM$ may be defined. Specifically, consider any vector $X \in T_xM$ and a frame q at x. The horizontal lift of X is then a unique horizontal vector $X^* \in H_qF(M)$ such that its projection returns the original vector itself, i.e. $\pi_* X^* = X$. Now consider any Euclidean vector $E \in \mathbb{R}^d$. The vector $H_E(q)$ at the point q in $F(M)$ is defined by the horizontal lift of the vector qE on M, i.e. $H_E(q) = (qE)^*$. Hence, $(qE)^*$ may be interpreted as a horizontal vector field on $F(M)$. Corresponding to the unit (orthonormal) coordinate vectors $E_1, ..., E_d$ in $\mathbb{R}^d$, we note that $H_i := H_{E_i}$, $i = 1, ..., d$, are the associated horizontal vector fields of the frame bundle that span the horizontal subspace $H_qF(M)$ at each $q \in F(M)$.

We may adopt any valid local chart $x = \{x^i\}$ in a neighbourhood $O \subset M$. Using the inverse of the projection map, this local chart on the base manifold M induces a local chart $\tilde{O} = \pi^{-1}(O)$ in $F(M)$. Thus, let $X_i = \frac{\partial}{\partial x^i}$, $1 \leq i \leq d$, be the coordinate basis vectors. For a frame $q \in \tilde{O}$, we have $qE_i = Q_i^j X_j$ for some matrix $Q = (Q_j^i)$. Accordingly, we get $(x, q) \in \mathbb{R}^{d+d^2}$ as the local chart for $\tilde{O}$. Then, the vertical subspace $V_qF(M)$ is spanned by $X_{kj} = \frac{\partial}{\partial Q_j^k}$, $1 \leq j, k \leq d$. Also, the vector fields $\{X_i, X_{ij}, 1 \leq i, j \leq d\}$ span $T_qF(M)$, $q \in \tilde{O}$. An expression for the horizontal vector field H_i in terms of the local coordinates is given as follows.

$$H_i(q) = Q_i^j X_j - Q_i^j Q_m^l \gamma_{jl}^k(x) X_{km} \tag{2.11}$$

For the sake of brevity, we skip the proof here and refer to ([23]).

From the definition of q_t, which is the horizontal lift of a smooth curve x_t on M, we have $q_t^{-1}\dot{x}_t \in \mathbb{R}^d$ since $\dot{x}_t \in T_{x_t}M$. We define the anti-development of $\{x_t\}$ on M as a curve u_t in $\mathbb{R}^d$ such that the following equation is satisfied.

$$u_t = \int_0^t q_s^{-1}\dot{x}_s ds.$$

In other words, $q_t \dot{u}_t = \dot{x}_t$ and by the definition of horizontal vector fields, we have $H_{\dot{u}_t}(q_t) = (q_t \dot{u}_t)^* = (\dot{x}_t)^* = \dot{q}_t$, i.e. the anti-development u_t and the horizontal lift q_t of a curve x_t on M are simply related by an ODE. In view of our work in the next subsection, it is expedient to rewrite the last equation as

$$\dot{q}_t = H_i(q_t)\dot{u}_t \tag{2.12}$$

If we start from an Euclidean curve u_t in $\mathbb{R}^d$ and a frame q_0 at the point x_0 on M, the unique solution of the above ODE is given by a horizontal curve q_t in $F(M)$. We refer to this horizontal curve as the development of u_t in the frame manifold $F(M)$. Its projection on M given by πq_t is called the development of u_t in M.

2.2.4 Local coordinate expression of a developed SDE on RM

We extend equation 2.12 to the stochastic case and write it in the Stratonovich sense as:

$$dq_t = H_i q(t) \circ dW_t^i \tag{2.13}$$

where the Ito SDE for the Euclidean stochastic process W_t has the following form:

$$dW_t^i = \alpha^i(W_t)dt + \beta_j^i(W_t)dB_t^j \tag{2.14}$$

From [23] (see proposition 2.1.3), the horizontal vector fields are locally given by the equation below.

$$H_i(q) = Q_j^i X_j - Q_j^i Q_m^l \gamma_{jl}^k X_{km} \tag{2.15}$$

where

$$X_i = \frac{\partial}{\partial x^i} \quad X_{km} = \frac{\partial}{\partial Q_m^k} \tag{2.16}$$

Hence, written in the Stratonovich sense, the equation for $q_t = \{x_t^i, Q_j^i(t)\}$ is

$$dx_t^i = Q_j^i(t) \circ dW_t^j \tag{2.17}$$

$$dQ_j^i(t) = -\gamma_{kl}^i(x_t) Q_j^l(t) Q_m^k(t) \circ dW_t^m \tag{2.18}$$

From equation (2.17) and in the Ito sense, we have

$$\begin{aligned} dx_t^i &= Q_j^i(t) dW_t^j + \frac{1}{2} d\langle Q_j^i(t), dW_t^j \rangle \qquad (2.19) \\ &= Q_j^i(t)\alpha^j(W_t)dt + Q_j^i(t)\beta_m^j(W_t)dB_t^m + \frac{1}{2} d\langle Q_j^i(t), dW_t^j \rangle \end{aligned}$$

Let $dM_t^i = Q_j^i(t)\beta_m^j(W_t)dB_t^m$ be the martingale part. Then we have

$$\begin{aligned} d\langle M_t^i, M_t^j \rangle &= \langle Q_m^i(t)\beta_r^m(W_t)dB_t^r, Q_n^j(t)\beta_s^n(W_t)dB_t^s \rangle \qquad (2.20) \\ &= Q_m^i(t)\beta_r^m(W_t)Q_n^j(t)\beta_r^n(W_t)dt \qquad (2.21) \end{aligned}$$

However, we have $qE_l = Q_l^i X_i$ and $\delta_{lm} = \langle qE_l, qE_m \rangle = \langle Q_l^p X_p, Q_m^q X_q \rangle = Q_l^p Q_m^q \langle X_p, X_q \rangle = g_{pq} Q_l^p Q_m^q$. Thus, $QgQ^T = I$ or $Q^T Q = g^{-1}$. Accordingly, we may write

$$d\langle M_t^i, M_t^j \rangle = [\beta^T g^{-1} \beta]^{ij} dt \tag{2.22}$$

Now, let $\sigma = Q\beta$. Note that the identity $Q = \sqrt{g^{-1}}$ is a direct consequence of the torsion-free nature of the frame-bundle, i.e. the orthonormal frame retains the orthonormality upon parallel transport along a smooth curve on the base manifold. It is equivalent to the requirement that the connection 1-forms in the cotangent space at any point on the base manifold are skew symmetric. Indeed, the same identity has been used in the literature while developing Brownian motion on an RM. We

may now write

$$dM_t^i = [Q\beta]_m^i dB_t^m \tag{2.23}$$

$$= \sigma_m^i dB_t^m \tag{2.24}$$

From equation (2.18), we have

$$dQ_j^i(t) = -\gamma_{kl}^i(x_t)Q_j^l(t)Q_m^k(t)dW_t^m + \frac{1}{2}d\langle -\gamma_{kl}^i(x_t)Q_j^l(t)Q_m^k(t), dW_t^m\rangle \tag{2.25}$$

Thus, the last term on the RHS of equation (2.19) becomes

$$\begin{aligned}
d\langle Q_j^i, dW_t^j\rangle &= \langle dQ_j^i, dW_t^j\rangle \qquad (2.26)\\
&= \langle -\gamma_{kl}^i(x_t)Q_j^l(t)Q_m^k(t)dW_t^m, dW_t^j\rangle\\
&= \langle -\gamma_{kl}^i(x_t)Q_j^l(t)Q_m^k(t)[\alpha^m(W_t)dt + \beta_p^m(W_t)dB_t^p], [\alpha^j(W_t)dt + \beta_q^j(W_t)dB_t^q]\rangle\\
&= -\gamma_{kl}^i(x_t)Q_j^l(t)Q_m^k(t)\beta_p^m\beta_q^j\langle dB_t^p, dB_t^q\rangle\\
&= -\gamma_{kl}^i(x_t)Q_j^l(t)Q_m^k(t)\beta_p^m\beta_p^j dt\\
&= -\gamma_{kl}^i(x_t)[Q\beta]_p^l[Q\beta]_p^k dt\\
&= -\gamma_{kl}^i(x_t)\sigma_p^l\sigma_p^k dt
\end{aligned}$$

Note that the contraction $\beta_p^m\beta_p^j$ over p makes sense as the diffusion coefficient β in equation (2.14) has the same representation in its covariant, contravariant or mixed forms. Substituting in equation (2.19), we arrive at the developed SDE.

$$dx_t^i = [\sqrt{g^{-1}(x_t)}]_j^i\alpha^j(W_t)dt + \sigma_m^i(W_t)dB_t^m - \frac{1}{2}\sigma_p^l\sigma_p^k\gamma_{kl}^i(x_t)dt \tag{2.27}$$

where $\sigma = Q\beta$ as mentioned earlier. Finally, we must emphasize the important role the Ito representation plays in incorporating the evolution of the frame field given by equation (2.25) within the local coordinate expression for the developed SDE as above. In particular, we do not need an additional SDE describing the evolving frame field. Without the Ito correction term in equation (2.19), this would not have been possible.

2.3 Riemannian manifold embedded in a Euclidean space: a special case

In order to glean additional insights into stochastic development, we consider the special case where a curve in $\mathbb{R}^d$ is required to be projected on an $d-1$-dimensional RM which can be embedded in $\mathbb{R}^d$. For further simplicity and ease of illustration, let $d = 3$ and the RM be a 2-dimensional surface S. Moreover, let us first consider the problem of deterministic orthogonal projection of a space curve (in $\mathbb{R}^3$) on the surface S. Let S be parametrized as:

$$S = \phi(u^1, u^2) : u^1, u^2 \in U \subset \mathbb{R}^2 \tag{2.28}$$

where ϕ may be regarded as a Euclidean 3-vector. Note that U is an open patch of parametric co-ordinates u^1, u^2. Let e_1, e_2 be the 3-dimensional basis vectors on this parametric patch.

Let x be a point on S. On the tangent space T_xM, define the tangent vectors $g_1 = \frac{\partial \phi}{\partial u^1}$ and $g_2 = \frac{\partial \phi}{\partial u^2}$ so that the Riemannian metric G is given in components by

$$G_{ij} = g_i \cdot g_j \tag{2.29}$$

Now, in line with the frame field approach we have adopted in Section 2.2.3 for stochastic development, let $F : T_xM \to \mathbb{R}^2$ be a frame at $x \in S$ such that

$$F_{ij} g_j = e_i \tag{2.30}$$

where e_i denotes the representation of the Euclidean vector e_i on T_xM. We may now write:

$$< e_i, e_j >= \delta_{ij} = F_{ik} G_{kl} F_{lj} \implies F^T F = G^{-1} \tag{2.31}$$

We thus have $F = \sqrt{G^{-1}}$ as a possible solution. At this stage, consider a space curve $p(t)$ in $\mathbb{R}^3$ and let $q(t) = q(u^1(t), u^2(t))$ be its orthogonal projection on S. Following [27], the basic idea is to find the point $q(t) \in S$ such that the Euclidean vector $\vec{qp}(t)$ is normal to the tangent space $T_{q(t)S}$ at $q(t)$. This requires that

$$\vec{qp}(t).g_i = 0 \ \ \forall i \tag{2.32}$$

where $\vec{qp}(t) = q(t) - p(t)$, so that upon differentiation with respect to t, we have

$$\frac{dq}{dt}.g_i + \vec{qp}(t).\frac{dg_i}{dt} = \frac{dp}{dt}.g_i \tag{2.33}$$

Using chain rule, viz. $\frac{dq}{dt} = \frac{dq}{du^i}\frac{du^i}{dt} = g_i \dot{u}^i$ and $\frac{dg_i}{dt} = \frac{\partial g_i}{\partial u^j}\dot{u}^j$, we arrive at

$$\left(g_i.g_j + \vec{qp}.\frac{\partial g_i}{\partial u^j} \right) \frac{du^j}{dt} = \frac{dp}{dt}.g_i \tag{2.34}$$

If n denotes the unit normal to T_xM at q, then $\vec{qp} = \rho n$ for a positive scalar ρ. Moreover, $\left\{ H_{ij} = \frac{\partial g_i}{\partial u_j}.n \right\}$ is the matrix of the second fundamental form. Thus, we may rewrite equation 2.34 as:

$$\left(G_{ij} + \rho H_{ij} \right) \dot{u}^j = \dot{p}.g_i \tag{2.35}$$

To see how equation 2.35 for the evolution of $\dot{u}^j(t)$ is related to the frame (that played a role in our intrinsic description of development in Section 2.2.3), we note that $H_{ij} = \frac{\partial g_i}{\partial u^j}.n$ contains the normal n. Indeed, the evolution of n in the ambient 3-dimensional Euclidean space along with the orthogonality of n with $g_i, i = 1,2$ determines how the frame bundle evolves in the present context. In other words, $n(t)$ should be treated as an additional variable (analogous to the frame evolution component of H_i in equation 2.27 Section 2.2.4) and its evolution may be determined

by rewriting equation 2.32 as:

$$\rho(t)n(t).g_i = 0 \tag{2.36}$$

and differentiating it with respect to t. Defining $K_{ij} = \frac{\partial g_i}{\partial u^j}$, which is a vector for each $(i,j)-$pair, we may rewrite equation 2.35 as

$$\left(G_{ij} + \rho(t)K_{ij}.n(t)\right)\dot{u}^j = \dot{p}.g_i \tag{2.37}$$

$$\implies \dot{u}^i = \left(\mathbb{I} + G^{-1}K.\int_0^t \frac{d}{ds}[\rho(s)n(s)]ds\right)^{-1}_{ik} G^{-1}_{kj}\dot{p}.g_j \tag{2.38}$$

$$= G^{-1}_{ij}\dot{p}.g_j + \text{h.o.t.} \tag{2.39}$$

Although the indicial notations for the matrix G are as per usual, there is perhaps a slight abuse of notation in those for the vector g_i's in the equation above. Typically, g_i would represent the $i-$th component of a vector g, but in the equation above, g_i is itself a vector for $i = [1,2]$. Upon an Euler discretization of the ODE in equation 2.39 and choosing $\rho(0) = 0$, we see that the second term on the RHS is of $O(\Delta t)^2$(Δt being the time step-size), and hence of higher order. Now, using $Fg_i = e_i$ or $g_i = F^{-1}_{ij}e_j = \left[\sqrt{G}\right]_{ij}e_j$, we may write

$$\dot{u}^i = G^{-1}_{ij}\dot{p}.g_j \tag{2.40}$$

$$= F^T_{ik}F_{kj}\dot{p}.F^{-1}_{jl}e_l \tag{2.41}$$

$$= F^T_{ik}\delta_{kl}\dot{p}.e_l \tag{2.42}$$

$$= F^T_{il}\dot{p}.e_l \tag{2.43}$$

$$= \left[\sqrt{G^{-1}}\right]_{il}\dot{p}.e_l \tag{2.44}$$

$$= \left[\sqrt{G^{-1}}\right]_{il}\dot{p}_l \tag{2.45}$$

This last equation is the same as the co-ordinate component of equation 2.12 in Section 2.2.4 (since the frame component is explicitly accounted for already), now arrived explicitly for a 2-dimensional surface embedded in 3-dimensional Euclidean space, unlike the intrinsic description using moving frames in Section 2.2.3. The

stochastic counterpart may then be derived in the same manner as in Section 2.2.4.

2.4 Illustrative examples

In this section, we consider the application of stochastic development to the minimization of a non-convex objective function without constraints. Within a stochastic search framework, we specifically do this by developing the overdamped Langevin SDE whose evolution is additionally guided by a simulated annealing procedure. In this context, note that a strictly positive, smooth, scalar-valued and non-convex objective function $f(x)$ could be looked upon, at least locally, as an energy-like functional in the space of the design variables. We consider the minimization of the Ackley and Rastrigin functions which constitute two of the benchmark problems in [28], often used to test the performance of an optimization scheme. Treating $f(x)$ as the energy, we may readily determine g and γ; see Appendix A for details. During a stochastic search involving a non-convex function, g may sometimes become negative-definite, particularly during the initial stages. As noted before, we use an additive regularizer of the type ΥI in order to ensure positive-definiteness of g. We then use the developed SDE for the overdamped Langevin dynamics with simulated annealing to carry out the evolutionary search for the global minimum of $f(x)$. The results so obtained are also contrasted with those via the overdamped Langevin dynamics with simulated annealing, but without stochastic development as well as the equation for Langevin diffusion on Riemannian manifolds from the literature. One may note that the simulated annealing step expedites a more exhaustive search of the design space during the initial stages.

2.4.1 The Ackley function

The Ackley function $f(x)$ to be minimized is given by:

$$f(\mathbf{x}) = f(x_1,...,x_n) = -a\exp\left(-b\sqrt{\frac{1}{n}\sum_{i=1}^{n}x_i^2}\right) - \exp\left(\frac{1}{n}\sum_{i=1}^{n}\cos(cx_i)\right) + a + \exp(1) \tag{2.46}$$

where n is the dimension of x, the design variable. The overdamped Langevin SDE is of the following form.

$$dX_t = -\beta_t \nabla f(X_t)dt + \sqrt{2\beta_t}dB_t \tag{2.47}$$

Its stochastically developed version as per equation 2.27 is:

$$dX_t = -\beta_t \sqrt{g^{-1}} \nabla f(X_t)dt + \sqrt{2\beta_t}\sqrt{g^{-1}}dB_t - \beta_t g^{-1}\gamma dt \tag{2.48}$$

where

$$g_{ij} = \frac{1}{2}\frac{\partial^2 f(\mathbf{x})}{\partial x_i \partial x_j} + \Upsilon \delta_{ij} \tag{2.49}$$

The equation existing for diffusions on RM (typically used for Markov chain Monte Carlo (MCMC)) is [29, 30, 31, 32, 33]

$$dX_t = -\beta_t g^{-1} \nabla f(X_t)dt + \sqrt{2\beta_t}\sqrt{g^{-1}}dB_t - \beta_t g^{-1}\gamma dt \tag{2.50}$$

The Riemannian connection γ can be obtained from the derivatives of g (Appendix A). Since we need to compute g and γ to arrive at the developed SDE, our scheme is not gradient-free unlike most others based on meta-heuristics, e.g. the genetic algorithm. However, when the gradient of the objective function is available, it is expected that the present approach should have the benefit of a relatively faster convergence. For a 40-dimensional Ackley function, we have reported the results in Figure 2.3. An ensemble size of only five particles has been used for this purpose. The problem parameters used are as provided in Table 2.1. As can be seen in the figure, the solutions through the Euclidean route and the existing equation for Langevin diffusion on an RM fail to converge for the 40-dimensional problem, while the stochastically developed version converges within 40 steps. We may note that the Euclidean version works for the 2-dimensional case; see Figure 2.2. However, in this case too, the quality of performance of the geometric version is much better. In reporting these results, the algorithm parameters are so chosen (by trial and error) as to represent the best performance of each method. More specifically,

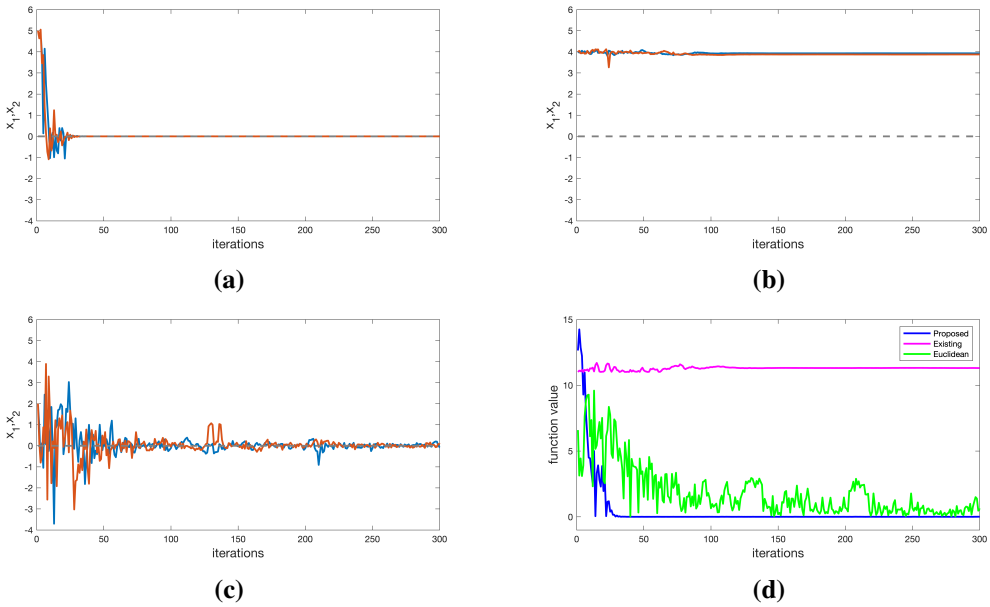


Figure 2.2: 2-dimensional Ackley function minimization: results by the proposed method are shown in (a) those via equation 2.50 in (b) and its Euclidean counterpart in (c). Evolutions of function values pertaining to these three methods are compared in (d).

the integration step size dt for the proposed scheme is always chosen about two orders of magnitude higher than its Euclidean counterpart. The annealing parameter β represents a temperature-like quantity in the diffusion equation, a higher value implying more exploration of the search space. This may also be perceived as scaling of the integration time step dt with time - which is why a smaller value is used in the Euclidean case than in the geometric case. For an optimal performance in terms of fastest convergence to the true function value (known for the benchmark problems considered), both the dt and β values have to be tuned for any method. Several pairs of these are tried for one method at a time, and the one resulting in the best (visually) function value plot is chosen.

2.4.2 The Rastrigin function

The Rastrigin function is given by the following expression:

$$f(\mathbf{x}) = f(x_1,...,x_d) = \sum_{i=1}^{d} [b(x_i)^2 - a\cos(2\pi x_i)] + a \times d \tag{2.51}$$

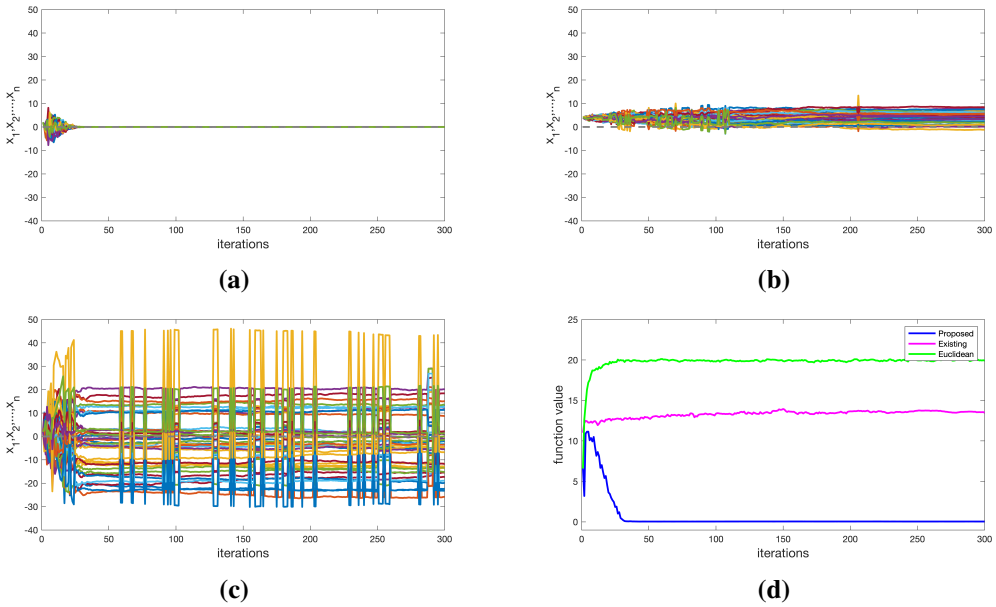


Figure 2.3: 40-dimensional Ackley function minimization: results by the proposed method are shown in (a) those via equation 2.50 in (b) and its Euclidean counterpart in (c). Evolutions of function values pertaining to these three methods are compared in (d).

Similar to the Ackley function, results for this case are provided in Figures 2.4 and 2.5. The results show a similar trend to that of the Ackley function. The parameters used for each of the methods are again provided in Table 2.1.

2.5 Concluding remarks

We have only provided an outline of what seems to be a potentially powerful and geometrically inspired stochastic search scheme, applications of which could be many. In the presence of an equality constraint, one may readily modify the energy function by adding a term that penalizes the violation of the constraint. While we have adopted an energy-based route for the stochastic search, a geometrically developed version of a martingale based approach [34] could as well be used. A geometric variant of the stochastic approximation framework [35] is also worth exploring. Note that, within the current setup, constrained optimization problems could also be solved through an appropriate modification of the energy, viz. via a penalty term similar to the first two problems. Indeed, given any cost function (path dependent or otherwise) and a diffusion process, one may construct an appropriate Feynman-Kac

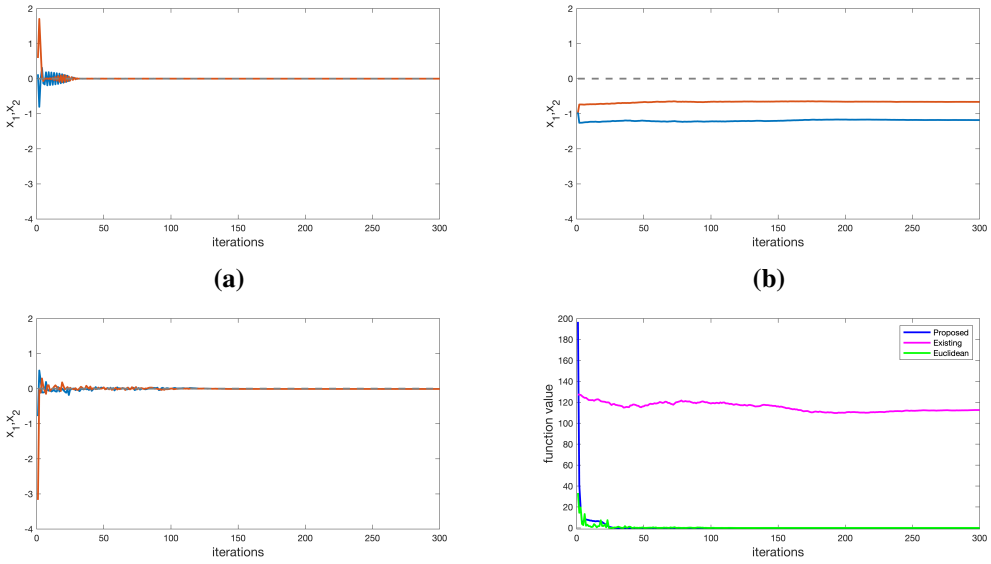


Figure 2.4: 2-dimensional Rastrigin function minimization: results by the proposed method are shown in (a) those via equation 2.50 in (b) and its Euclidean counterpart in (c). Evolutions of function values pertaining to these three methods are compared in (d).

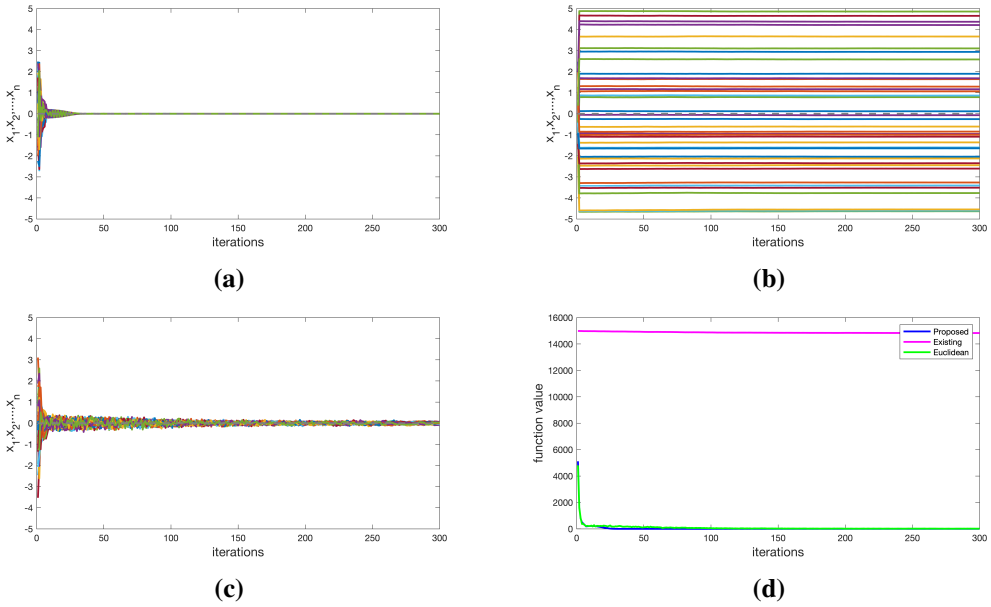


Figure 2.5: 40-dimensional Rastrigin function minimization: results by the proposed method are shown in (a) those via equation 2.50 in (b) and its Euclidean counterpart in (c). Evolutions of function values pertaining to these three methods are compared in (d).

	Proposed	**Existing**	**Euclidean**
2-dimensional Ackley			
step size	0.5	0.05	0.01
β_0	1000	1000	50
decay factor ν	0.01	0.001	0.001
regularizer Υ	10^6	10^6	-
40-dimensional Ackley			
step size	0.5	0.5	0.001
β_0	50000	50000	1000
decay factor ν	0.01	0.0001	0.01
regularizer Υ	10^6	10^6	-
2-dimensional Rastrigin			
step size	0.1	0.1	0.001
β_0	50	50	5
decay factor ν	0.001	0.0001	0.001
regularizer Υ	10^6	10^6	-
40-dimensional Rastrigin			
step size	0.1	0.01	0.001
β_0	50	50	5
decay factor ν	0.001	0.0001	0.0001
regularizer Υ	10^6	10^6	-

Table 2.1: Parameters used for the three equations used for optimization for all the problems considered, the Ackley and Rastrigin functions in 2 and 40 dimensions. Step size is dt. β_0 is the initial value of β in equations 2.47 (proposed), 2.50 (existing) and 2.48 (Euclidean). β is decayed with time as per the following expression: $\beta_{k+1} = \frac{\beta_k}{\exp(\nu)k}$ and Υ is the regularizer used for the metric in equation 2.49

path integral from which an action functional could be identified. This action functional could offer a general approach to construct a Riemannian metric and hence stochastically develop the underlying diffusion process. We intend to consider these issues in the future.

In the next chapter, we discuss the application of the main result derived in this chapter, i.e. equation 2.27 to MCMC. The methodology is essentially the same as that adopted for optimization, except for a few details. One, annealing is not made use of, i.e. $\beta = 0.5$ in equation 2.48 at all time steps. Instead, a Metropolis-Hasatings accept-reject step is used. Two, the Fisher information matrix (FIM) is used as the Riemannian metric. We also discuss why the FIM is an appropriate metric to use for statistical parameter estimation problems.

Chapter 3

Stochastically developed Langevin dynamics: application to Markov chain Monte Carlo

This chapter is a modified version of the work in [36], a preprint. The section containing a proof in the preprint (an alternative to the proofs provided in sections 2.2 and 2.3 of Chapter 1) is moved to Appendix A of this thesis for a better flow of the presentation.

3.1 Introduction

Markov Chain Monte Carlo (MCMC) is an active field of research with a rich body of literature that is fast-growing. Significant applications of an MCMC algorithm include, among others, evaluating a complex integral and sampling from an unnormalized distribution. The latter is especially useful when it is difficult to obtain the normalizing constant of a distribution or when sampling from the density is quite non-trivial even though the density may itself have a simple form. MCMC is perhaps the only known general approach to find the volume enclosed by an n-dimensional convex body with a reasonable computational overhead [37]. It has also been used to sample from the posterior probability in stochastic filtering problems based on Sequential Monte Carlo (SMC). In a more general context, MCMC has been employed for optimization as well, see e.g. [38]. In [39], the authors

discuss and demonstrate an interesting approach to modify a few existing MCMC methods to enable their applications to a wide range of problems including but not limited to data assimiliation and image registration. MCMC methods, in combination with existing machine learning algorithms, have also been exploited in applications such as particle filtering [40], robotics [41], computational biology [42], genetics [43] and machine learning [44], to name only a few.

We refer to [45] for a recent review of MCMC methods, with an interesting discussion on a few popular misconceptions. The Metropolis Adjusted Langevin Algorithm (MALA) [4], the Hamiltonian Monte Carlo (HMC) approach [46] and related methods that make use of the gradient of the target density to design a proposal distribution for the Markov chain may be considered as 'first-order'. MALA uses Langevin dynamics in conjunction with the Metropolis accept-reject step. There are several MCMC algorithms that are based on Langevin dynamics, e.g. the Metropolis adjusted Langevin truncated algorithm or MALTA [4], the projected unadjusted Langevin algorithm (ULA) [47], proximal MALA [48], underdamped Langevin MCMC [49], Moreau-Yosida Unadjusted Langevin Algorithm (MYULA) and Moreau-Yosida Regularized Metropolis Adjusted Langevin Algorithm (MY-MALA) [50], the ULA [2] which is free from the Metropolis accept-reject step and is essentially the approach used in Chapter 2. Note that [4], [51], [52], [53], [2], [54], investigated the convergence properties of various Langevin diffusion based MCMC methods. There are also several studies that focus on the scaling, convergence and mixing properties of the Langevin-class of MCMC algorithms. For instance, in the context of sampling from a log-concave density using MALA, [55] proves a non-asymptotic upper bound on the mixing time to demonstrate the benefit of the accept-reject step, viz. an exponentially improved dependence on error tolerance. Similar bounds on the error of sampling from a target density based on three different schemes of discretized Langevin dynamics have been reported [56]. [57] proposes a new approach to quantify convergence of underdamped Langevin dynamics to equilibrium.

In this work, our focus is on MALA. It is a class of MCMC methods in which

the Markov Chain evolves as per the overdamped Langevin dynamics. Specifically, the Langevin SDE (interpreted in the sense of Ito) is given by

$$dx_t = \frac{1}{2}\nabla L dt + dB_t. \tag{3.1}$$

Here L is the log-likelihood of the target density and dB the standard Brownian increment such that $dB \sim \mathcal{N}(0, \sqrt{dt})$. Since the Langevin dynamics involves gradient information of the target distribution, the method is more likely to move towards regions of high probability, which is a major advantage over the use of largely arbitrary proposal distributions.

If the Langevin SDE could be solved exactly, all the particles would be accepted and there would be no need for a Metropolis adjust step. However this is rarely the case. The SDE may be solved by various numerical integrators – the Euler-Maruyama method being often used – that introduce integration errors and hence necessitates the Metropolis-Hastings accept-reject step. This step also helps improving the convergence characteristics of the algorithm. However, MALA does have its share of disadvantages, e.g. cases involving highly correlated multivariate distributions.

As we have just noted, MALA is based on the Euclidean Langevin SDE. Working with SDEs in the Euclidean setting however comes with its shortcomings. The two major inadequacies of working in the Euclidean setting are as follows. First, owing to the noise term in the SDE, there is a possibility of a gradual increase in the variance of numerical solutions to SDEs. Second, the space-filling properties of Brownian motion may cause delayed convergence. These issues gain in importance as the dimension of the problem increases and form the motivation for our present work. Despite the spectrum of research areas in which MCMC finds application and the many flavours of it that have been explored, hardly an effort has been made at exploiting the differential geometric aspects to develop faster and more accurate algorithms. Whenever diffusions on Riemannian manifolds (RM) are considered, it is either directly in the language of frame bundles or exponential and log maps which is inaccessible to non-specialists, see e.g. [58], or in the form of an SDE which

simultaneously uses Amari's natural gradient [59] in conjunction with the equation for Brownian motion on an RM. [33] is a work on MCMC belonging to the latter category. To our understanding, this work is however beset with certain issues (discussed in detail in Section 3.4) which is indeed one of the motivating factors for this article. We present here an MCMC method based on the stochastic development of a general SDE on an RM as derived in the previous chapter and use it specifically in the context of the overdamped Langevin diffusion equation to obtain the geometrically adapted version of MALA, which we will refer to as Geometrically Adapted Langevin Algorithm (GALA) from here on. The resulting algorithm for GALA is also given in Section 3.2, which may be considered a 'second-order' method, as it makes use of derivatives up to the second order for the proposal step; this is unlike MALA which is a 'first-order' method.

The rest of the chapter is organized as follows. In Section 3.2, we derive the stochastic development of an SDE on the RM starting with a brief review of differential geometry and stochastic calculus for completeness. Section 3.3 discusses the relevance of Fisher-Information matrix as a Riemannian metric for the manifold MCMC methods. Section 3.4 discusses work related to GALA. Section 3.5 contains an illustration of the method on the couple of problems discussed above. We conclude the chapter in Section 3.6 with a discussion on the outcomes and an outline of the future scope.

3.2 Methodology

We aim at estimating the true parameter vector θ^* of a probability density given a set of samples drawn from it. With our interest in MALA, the evolution of the parameter vector $\theta(t)$ is governed by the Langevin SDE,

$$d\theta(t) = \frac{1}{2}\nabla L(\theta(t))dt + dB_t \tag{3.2}$$

where L is the log likelihood. In accordance with (2.27), the stochastically developed counterpart of (3.2) is then given by

$$d\theta_t^i = \frac{1}{2}[\sqrt{g^{-1}(\theta_t)}]_{ij}\nabla L(\theta_t)^j dt + [\sqrt{g^{-1}(\theta_t)}]_{im} dB_t^m - \frac{1}{2}[g^{-1}(\theta_t)]_{kl}\gamma_{kl}^i(\theta_t)dt \quad (3.3)$$

Algorithm for GALA

While the pseudo-code presented below could be readily adapted for problems related to sampling from given densities, it is presently for purposes of estimating the parameter vector θ^* of a given distribution using GALA, when the observations $\{z\}_{i=1}^N$ are available from a known probability density function $p_x(.;\theta^*)$, where $x \in \Omega$ and $(\Omega, \mathscr{F}, \mathscr{P})$ is a complete probability space.

Algorithm 1: GALA

Result: MCMC chain of length K
Data: N samples $\{z\}_{i=1}^N$ distributed as per $p_x(.;\theta^*)$
Initialize: θ_τ for $\tau = 1$
for $\tau = 1 : K-1$ **do**
  Evaluate the log-likelihood $L(\theta_\tau)$ as $L(\theta_\tau) = \log(p_x(z|\theta_\tau))$
  where $p_x(z|\theta_\tau) = \prod_{i=1}^N p_x(z_i;\theta_\tau)$
  Obtain the Fisher-Information matrix (FIM), which is the Riemannian metric $g(\theta_\tau) = E[(\nabla_\theta L(\theta_\tau))(\nabla_\theta L(\theta_\tau))^T]$
  Determine the Riemannian connection $\Gamma_{ij}^k(\theta_\tau) = \frac{1}{2}g^{kl}(\theta_\tau)[\partial_i g_{jl}(\theta_\tau) + \partial_j g_{il}(\theta_\tau) - \partial_l g_{ij}(\theta_\tau)]$
  Integrating the SDE (3.3) by Euler-Maruyama method, we have the following proposal
  $\theta_{\tau+1}^i = \theta_\tau^i + [\sqrt{g^{-1}(\theta_\tau)}]_{ij}\nabla L(\theta_\tau)^j \Delta t + [\sqrt{g^{-1}(\theta_\tau)}]_{im} dB_t^m - \frac{1}{2}[g^{-1}(\theta_\tau)]_{kl}\Gamma_{kl}^i(\theta_\tau)\Delta t$
  Accept $\theta_{\tau+1}$ as per the Metropolis-Hastings acceptance probability α determined as follows:
  $\alpha = \min\left(\frac{p_x(z|\theta_{\tau+1})q(\theta_\tau|\theta_{\tau+1})}{p_x(z|\theta_\tau)q(\theta_{\tau+1}|\theta_\tau)}, 1\right)$
  where $q(y|x)$ is the multivariate Gaussian density with mean $\mu^i : x^i + [\sqrt{g^{-1}(x)}]_{ij}\nabla L(x)^j \Delta t - \frac{1}{2}[g^{-1}(x)]_{kl}\Gamma_{kl}^i(x)\Delta t$, covariance matrix $\Sigma_{im} : \Delta t[g^{-1}(x)]_{im}$ evaluated at y.
end
return $\{\theta_\tau\}_{\tau=1}^K$

3.3 Significance of Fisher-Information in statistical parameter estimation

The class of problems considered in this chapter comprise of estimating parameters of a certain probability distribution given independent and identically distributed (i.i.d.) samples from that distribution. The Riemannian metric used in these examples in equation 3.3 is the FIM. In this section, we try to glean insights into why this is an appropriate metric by making use of the Kullback-Leibler divergence [60].

Let N i.i.d. samples $Z = [z_i]_{i=1}^{N}$, be available from a distribution whose corresponding probability density is $p_{\theta^*}(x;\theta^*)$ where θ^* is unknown. The aim is to solve for the parameter θ^*. Digressing a bit from the main point to appreciate a couple of interesting points; we may also look at this problem as one maximising the likelihood $L(\theta;Z) = \Pi_{i=1}^{N} p_\theta(z_i;\theta)$ or the log-likelihood $\log L(\theta;Z)$ over θ. It is worth noting here that different realizations of Z may lead to different solutions of θ, however as $N \to \infty$, they converge in distribution to $\mathcal{N}(\theta^*, I(\theta^*)^{-1/2})$, see Chapter 6 in [61]. Unlike maximising the likelihood, when looked upon as an MCMC problem, we naturally obtain a distribution of the parameter estimate. Indeed, in the 1945 article by C R Rao, it is shown that the variance of the estimator is upper bound by the inverse of the FIM. The paper also discusses how the parametric space of probability densities can be treated as an RM. Coming back to the rationale of using the FIM as a Riemannian metric, consider the likelihoods corresponding to two parameter values θ^* and $\theta^* + \Delta\theta$. The Kullback-Leibler divergence between these likelihoods, where the one with θ^* is taken as the reference is given as follows

$$KL(L(\theta^* + \Delta\theta;Z)||L(\theta^*;Z)) \tag{3.4}$$

$$= \int \log\left(\frac{L(\theta^* + \Delta\theta;Z)}{L(\theta^*;Z)}\right) L(\theta^* + \Delta\theta;Z)dZ \tag{3.5}$$

$$= \int [\log L(\theta^* + \Delta\theta;Z) - \log L(\theta^*;Z)] L(\theta^* + \Delta\theta;Z)dZ \tag{3.6}$$

Expanding $\log L(\theta^*;Z)$ according to Taylor series about $\theta^*+\Delta\theta$, we have

$$\log L(\theta^*;Z) = \log L(\theta^*+\Delta\theta;Z) - \Delta\theta^T \nabla_{\theta^*+\Delta\theta} \log L(\theta^*+\Delta\theta;Z) \tag{3.7}$$

$$+ \Delta\theta^T \nabla^2_{\theta^*+\Delta\theta} \log L(\theta^*+\Delta\theta;Z)\Delta\theta \tag{3.8}$$

where $\nabla_{\theta^*+\Delta\theta} \log L(\theta^*+\Delta\theta;Z)$ and $\nabla^2_{\theta^*+\Delta\theta} \log L(\theta^*+\Delta\theta;Z)$ are the gradient and Hessian of the log-likelihood, respectively. Substituting the above expression in 3.6, we have

$$KL(L(\theta^*+\Delta\theta;Z)||L(\theta^*;Z)) \tag{3.9}$$

$$= \int [\Delta\theta^T \nabla_{\theta^*+\Delta\theta} \log L(\theta^*+\Delta\theta;Z) - \Delta\theta^T \nabla^2_{\theta^*+\Delta\theta} \log L(\theta^*+\Delta\theta;Z)\Delta\theta] L(\theta^*+\Delta\theta;Z) dZ$$

$$= \Delta\theta^T \mathbb{E}(\nabla_{\theta^*+\Delta\theta} \log L(\theta^*+\Delta\theta;Z)) - \Delta\theta^T \mathbb{E}(\nabla^2_{\theta^*+\Delta\theta} \log L(\theta^*+\Delta\theta;Z))\Delta\theta \tag{3.10}$$

where $\mathbb{E}(.)$ is the expectation with respect to the density $L(\theta^*+\Delta\theta)$. In what follows, we show that the expectation of the gradient of the log-likelihood vanishes.

$$\mathbb{E}(\nabla_{\theta^*+\Delta\theta} \log L(\theta^*+\Delta\theta;Z)) \tag{3.11}$$

$$= \mathbb{E}(\nabla_{\theta^*+\Delta\theta} \sum_{i=1}^{N} \log p(z_i;\theta^*+\Delta\theta)) \tag{3.12}$$

$$= \sum_{i=1}^{N} \mathbb{E}\left(\nabla_{\theta^*+\Delta\theta} \log p(z_i;\theta^*+\Delta\theta)\right) \tag{3.13}$$

$$= \sum_{i=1}^{N} \mathbb{E}\left(\frac{\nabla_{\theta^*+\Delta\theta} p(z_i;\theta^*+\Delta\theta)}{p(z_i;\theta^*+\Delta\theta)}\right) \tag{3.14}$$

$$= \sum_{i=1}^{N} \int \left(\nabla_{\theta^*+\Delta\theta} p(z_i;\theta^*+\Delta\theta)\right) dz_i \tag{3.15}$$

$$= \sum_{i=1}^{N} \nabla_{\theta^*+\Delta\theta} \int p(z_i;\theta^*+\Delta\theta) dz_i \tag{3.16}$$

$$= 0 \quad \because \int p(z_i;\theta^*+\Delta\theta) dz_i = 1 \forall i \tag{3.17}$$

Substituting $\mathbb{E}(\nabla_{\theta^*+\Delta\theta} \log L(\theta^*+\Delta\theta;Z)) = 0$ in equation 3.10, we have

$$KL(L(\theta^*+\Delta\theta;Z)||L(\theta^*;Z)) = -\Delta\theta^T \mathbb{E}(\nabla^2_{\theta^*+\Delta\theta} \log L(\theta^*+\Delta\theta;Z))\Delta\theta \quad (3.18)$$

Now, $-\mathbb{E}(\nabla^2_{\theta^*+\Delta\theta} \log L(\theta^*+\Delta\theta;Z))$, on the right-hand side of the equation above, is the FIM. We have thus shown that the error in distance in distributions due to an error of $\Delta\theta$ in the parameters is the norm of the error vector $\Delta\theta$ with respect to the FIM. Since we are interested in minimizing the distance in probability (corresponding to a given parameter value) with respect to the true density (corresponding to the true parameter value) rather than the Euclidean distance between two given parameter vectors, it makes sense to use the FIM as the Riemannian metric.

3.4 Related work

In this section, we discuss work related to GALA. The closest by far is [33]. The authors therein propose two major categories of MCMC methods on an RM, the first is based on Langevin dynamics and the second on Hamiltonian dynamics. Within the Langevin dynamics based methods, there are again two versions - the manifold Metropolis adjusted Langevin algorithm (MMALA) and the simplified MMALA. First, consider the MMALA which is closest to GALA. The equation for generating samples in [33] is proposed as (after correcting for a factor half in the last term):

$$\theta^i_{\tau+1} = \theta^i_\tau + [g^{-1}(\theta_\tau)]_{ij} \nabla L(\theta_\tau)^j \Delta t + [\sqrt{g^{-1}(\theta_\tau)}]_{im} dB^m_t - \frac{1}{2}[g^{-1}(\theta_\tau)]_{kl} \Gamma^i_{kl}(\theta_\tau)\Delta t \quad (3.19)$$

The authors use Amari's natural gradient ($[g^{-1}(\theta_\tau)]_{ij} \nabla L(\theta_\tau)^j$) in the equation for Brownian motion on an RM - a well-known equation in the literature. A rigorous justification of this step is however not available. While Amari's natural gradient is known to be a valid gradient of deterministic curves on an RM, it may not necessarily hold for diffusions. The article by Amari on natural gradient [59] has in fact used this gradient only for a deterministic method. Indeed, the origin of G^{-1} as a multiplying factor to the drift vector field appearing in a differential equation on an RM can be traced to certain basic principles of geometric mechanics; e.g. see

[62]. Specifically, a differential equation representing a balance law (e.g. of linear momentum) is essentially a balance of forces which in turn are co-vectors. Representing such an equation in terms of vectors (e.g. velocity, acc. etc.) requires the sharpening operation using G^{-1}; see Chapter 3 of [63]. Unfortunately, for SDEs written in terms of incremental states (co-vectors), a vector representation may not be not meaningful. This calls into question the use of G^{-1} as a multiplier of the drift.

In the simplified MMALA, the connection term (last term in (3.19)) is dropped, perhaps for the sake of simplification. In this setting, the invariant distribution of the proposal equation may not remain unchanged owing to the following reasons. First, since the SDE is different, so is the proposal density and its associated invariant distribution. The acceptance step may however enforce convergence to the target distribution, at least in some cases. Complications may arise, for instance, if the proposal SDE corresponds to a density whose invariant measure differs considerably from the target measure; it may no longer behave as an importance sampling scheme. In the Euclidean setting, the invariant distribution of any SDE pertains to the stationary solution of the Fokker-Planck equation. Hence, the case wherein simplified MMALA surely converges to the correct target distribution is when G is constant (see (B.3) in Appendix). This also corresponds to the preconditioned MALA, which is a well-established Euclidean MCMC method, though not an RM method. In practice, for parameter estimation problems, when enough data is available so that the posterior is almost a Dirac measure and as parameters converge to one value, simplified MMALA may behave as a preconditioned MALA as iterations progress. This may no longer hold when considering sampling problems when $G(X_t)$ never converges to one value, or even for parameter estimation problems when the data is scarce and the posterior has a large variance. An alternative might perhaps be to consider the multiplicative noise along with an appropriately added drift for the Langevin dynamics (see [64]) which is known to converge to the correct distribution. This last version has been arrived at in a follow-up article by [65] by a slightly different approach. It is done with the objective of correcting

the MMALA proposal in equation (3.19) so that it converges to the target distribution. In order to achieve this, a couple of adjustments are made to (3.19) so that it becomes equal to the multiplicatively driven Langevin system [64]. Although this proposal converges to the correct target distribution, it essentially remains a Euclidean method and not quite an RM proposal; see again [64].

Moving on to the RM-HMC method, the formalism in [33] seems to have been based on Theorem 6.4 of [66] which corresponds to the case when the Hamiltonian consists of only kinetic energy. This special case of Hamiltonian requires the evolution of the system state ϕ to follow a geodesic on the RM (so that $\nabla_{\dot{\phi}}\dot{\phi} = 0$). The Hamiltonian considered by [33] however consists of both potential energy and kinetic energy both, so the equations of motion presented need more justification. Intuitively, we can see that these may not be valid from the following perspective. For an n-dimensional manifold with coordinates $x^1, x^2 ... x^n$, the time derivative (velocity $\dot{x}^i$) is a tangent space object, see for instance Chapter 1 of [67]. The definition of acceleration would therefore necessitate the underlying Riemannian connection in the expression, since it requires an evolution across different tangent spaces. In other words, the time derivatives of momenta in Hamilton's equations must contain the Riemannian connection, see for instance the original work done in [68] where the Riemannian connection was duly incorporated. A more general case of this (presence of external forcing) is also derived in [69] and [62] by taking a variation of the action functional in terms of the Lagrangian, wherein the equations of motion are identical with those obtained in [68] for the case of no external forcing. Moreover, even though the phase space volume interpreted in the Euclidean space is conserved in [33], the Riemannian volume over the same phase space may not (e.g. see Chapter 3 of [63] for the computation of the Riemannian volume).

Unfortunately, in most of the available literature on RM based MCMC methods [70, 65, 71], the invariant distribution of a $d-$ dimensional proposal on an RM is examined using the Fokker-Planck equation evolving in a $d-$dimensional Euclidean space. We think this is inappropriate because any path evolving on an RM of dimension d is actually a $D-$dimensional path in the embedding Euclidean space,

where $D \geq d+1$ according to Nash's embedding theorem [72]. Therefore, if the invariant distribution were to be found using the Fokker-Planck equation, it should be accomplished in a Euclidean space of appropriately higher dimension. This is very challenging for the following reasons. The first is, in more cases than otherwise, it is difficult if not impossible to determine the embedding dimension D. Even if D can be determined, we must then re-write the $d-$ dimensional stochastically developed equation in the $D-$ dimensional (Euclidean) embedding space, which is often infeasible or difficult whilst defeating the very purpose of stochastic development. Finally, we would also need to redefine the probability space pertaining to the dimension D of the embedding space.

GALA does not satisfy the Euclidean Fokker-Planck equation, as expected, and is still successful for all the problems considered. A tempting possibility would perhaps be to consider the so-called Fokker-Plack equation on RMs [73]. GALA does not conform to this either, even though MMALA does (see (B.6) in Appendix). But, as numerically evidenced in the observed divergence of MMALA for most problems considered here, one anticipates that the last cited form of Fokker-Planck equation is perhaps not the right equation to study invariant distributions on RMs. Overall, the global properties of diffusions on RMs are far from adequately understood in the literature, though some first steps are taken. For instance, some work on the short time aymptotics of the heat kernel, which is related to the transition probability of a Brownian motion on an RM, has been reported in Chapters 4 and 5 in [23]. However, a more complete understanding of invariance may require an understanding of the long-term asymptotics, which may not be well-defined depending on the structure of the curvature tensor. Accordingly, the question of invariance, though important, remains unresolved as yet.

So far, the discussion was about the theoretical issues arising in the related work. The numerical examples reinforce these observations, wherein it is shown that MMALA fails to converge for all problems except the logistic regression case. Unlike in other cases, the metric for the logistic regression problem is symmetric for which the MMALA proposal via equation 3.19 reduces to the multiplicative

Langevin dynamics (see equation 2.50 in Chapter 2), which by construction [64] is a Euclidean proposal and converges to the target distribution. Even as the MMALA does converge for this problem, it takes longer compared to GALA as reported in Figure 3.8 and Table 3, thus reinforcing the argument of it being a Euclidean method. This contrast becomes more pronounced with an increase in the dimension.

Finally, a word about the problems considered in [33]; these are problems in which, typically, the connection term is either small or vanishes altogether. Therefore, in such problems, the method either reduces to or asymptotically behaves like pre-conditioned MALA. Owing to the symmetry in the metric, the logistic regression problem in particular, as mentioned in the previous paragraph, reduces to the multiplicative Langevin dynamics. However, for even a 1D problem when the data size is large or a high-dimensional correlated problem like the Gaussian example with unknown mean and covariance considered in this work, when the connection term becomes important in the proposal step, MMALA diverges as seen in Figures 3.1 and 3.5.

3.5 Illustrative examples

In this section, we illustrate the workings of GALA on a suite of related methods for two classes of parameter estimation problems. In the first, given a set of realizations from a probability distribution with a known functional form, we estimate the unknown parameters. The second problem concerns logistic regression wherein N number of explanatory variables with the corresponding binary response variables are given, and the aim is to reconstruct the regression parameters. Some of these problems have been considered in [33], though the authors therein work with one-dimensional Gaussian or uncorrelated multivariate Gaussian distributions. Extending the geometric construction from one to multivariate densities is however non-trivial, and this is what we accomplish in this section. In addition, we also consider the parameter estimation of Rayleigh, Weibull and Banana-shaped distributions by way of highlighting how an erroneous departure from proper stochastic development could either yield an incorrect solution or failure of the method for

the estimation problem involving a non-Gaussian density. We compare the results obtained with GALA with those obtained by MALA, MMALA, and in some cases corrected MMALA, No-U-turn sampler (NUTS) and HMC methods as well.

We consider toy problems ranging from a 1-dimensional Rayleigh to a 65-dimensional multivariate Gaussian and the heavy-tailed Weibull distribution to a highly twisted Banana-shaped distribution. The reason to work with such toy problems is to demonstrate the accuracy of estimation. As will be seen in the results for large dimensional examples, Stan [74, 75] and other methods converge to incorrect parameters. It is not too difficult to oversee this issue in problems with real datasets where the true parameter values remain unknown. The priors for all examples except for logistic regression is taken as uniform, as it slightly increases the problem difficulty and perhaps also leads to a fairer comparison among various methods. The initialization for various methods is kept the same except for Stan in which case its defaults are used. Figures 3.1-3.8 show the behaviour of various methods in the burn-in phase, which helps to visualize the speed of convergence. Tables display a comparison of various performance metrics of all the methods after the burn-in phase. Effective sample size (ESS) is often used as a performance metric which is fine for a sampling problem, since indeed samples are desired from a distribution. However, we believe it is perhaps not the right metric for parameter estimation problems considered in this work for the following reason - assuming enough data is available, the posterior distribution would be almost like a Dirac measure at the correct parameter value and an ideal algorithm should converge to the correct parameter value and stay there. However, ESS for such a solution would vanish, which is clearly not the right inference. Therefore, we do not consider ESS comparison, but instead use burn-in and acceptance rate, since a longer burn-in and a high rejection rate lead to wasted computation. We thus choose the algorithm parameters so as to reduce the burn-in period. The burn-in is determined based on the first time the Markov chain enters within a tolerance level of the correct parameter value and stays there. The acceptance percentage represents the number of samples accepted for the entire chain, i.e. including the burn-in phase. The estimated mean

and sample variance for the parameters are determined based on a certain number (different for different problems, as specified in the caption of each table) of samples after burn-in, while the true parameter value for each problem is mentioned in the black bar in the tables. For multi-variate cases, the norm of the sample variance (in Table 3.3) corresponds to the norm calculated from the sample variances of the scalar-valued components.

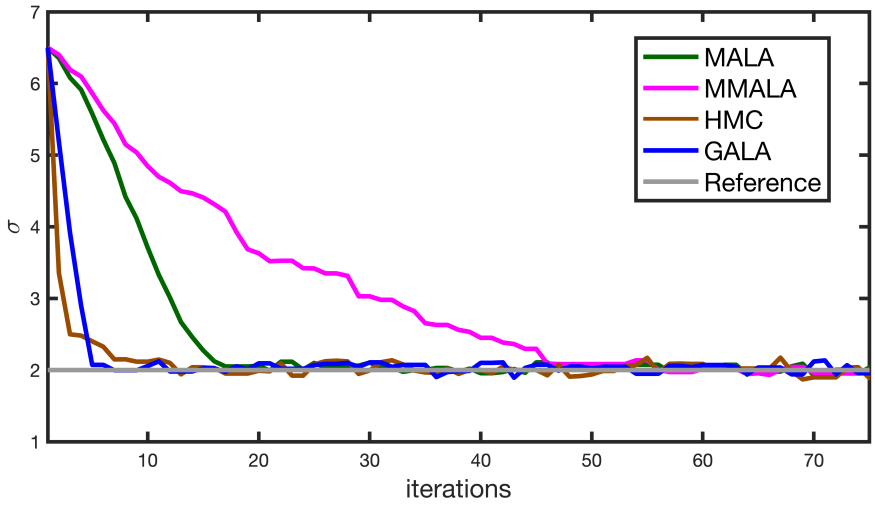


Figure 3.1: Parameter σ in the Rayleigh density via GALA ($\Delta t = 0.2$), MALA ($\Delta t = 0.01$), MMALA ($\Delta t = 0.15$) and HMC ($\Delta t = 0.04, L = 50$) for $N = 200$ sample observations

3.5.1 Estimating the parameters of a probability distribution

Rayleigh distribution. Consider a problem where N samples $\{z\}_{i=1}^{N}$ are available from a Rayleigh distribution with unknown parameter σ, which we wish to estimate. We first derive the developed equation for the Rayleigh distribution following the steps listed in the pseudo-code in Section 3.2 (see Appendix B for a detailed derivation)

$$d\sigma_t = \left(-\frac{\sqrt{N}}{2} + \frac{\sum_{i=1}^{N} z_i^2}{4\sigma^2\sqrt{N}} + \frac{\sigma_t}{4N} \right) dt + \frac{\sigma_t}{2\sqrt{N}} dB_t \tag{3.20}$$

which may be contrasted with those in MALA and MMALA as

$$\text{MALA:}\quad d\sigma_t = \left(-\frac{N}{\sigma} + \frac{\sum_{i=1}^{N}}{z_i^2}\right) dt + dB_t \tag{3.21}$$

$$\text{MMALA:}\quad d\sigma_t = \left(-\frac{\sigma_t}{4} + \frac{\sum_{i=1}^{N} z_i^2}{8\sigma_t N} + \frac{\sigma_t}{4N}\right) dt + \frac{\sigma_t}{2\sqrt{N}} dB_t \tag{3.22}$$

Results in the burn-in phase for parameter reconstruction by various methods are shown in Figure 3.1. Several performance metrics over 10 repeated simulations are summarized in Table 3.1. HMC requires integration over 50 steps of Hamiltonian dynamics for one proposal. This implies a similarly enhanced cost of proposal as shown in Table 3.1 for comparison. The sample variance (variance of the MCMC samples after discarding burn-in) obtained by GALA is lower compared to other methods.

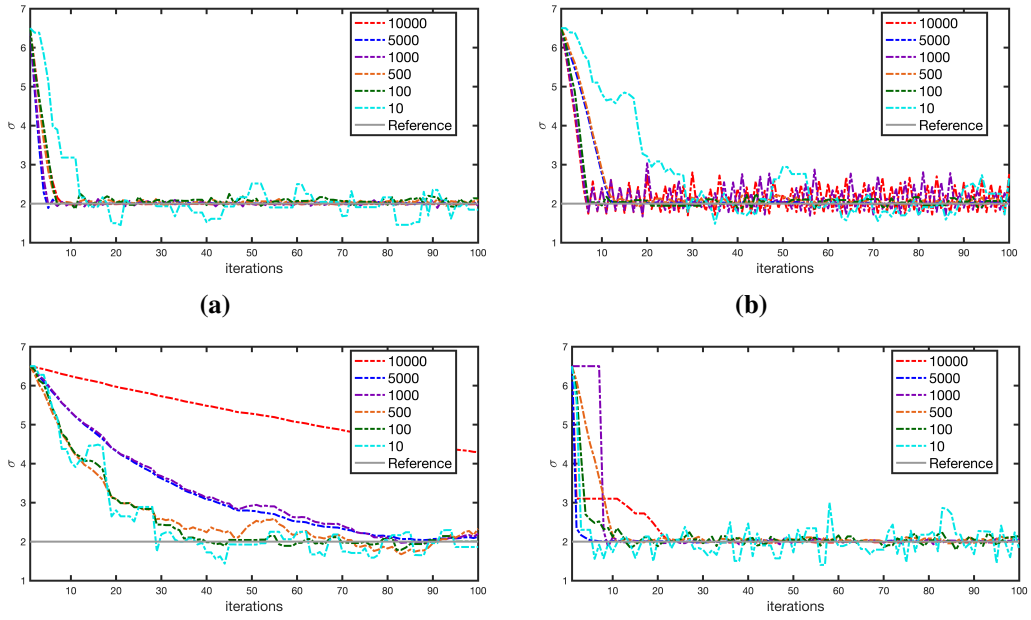


Figure 3.2: Reconstructing the parameter in the Rayleigh density: a comparison of results via different methods with varying number of observations; (a) GALA; (b) MALA; (c) MMALA; (d) HMC

As part of our convergence study, we now compare the performances of different methods as the number of observations increases. We present the results for Rayleigh distribution which, though one-dimensional, is unsymmetric enough to be

a good test problem. As anticipated and as shown in Figure 3.2, the performance of most methods improves, such as in the forms of ESS being higher and burn-in period smaller, with increasing number of observations. The exception is MMALA where it sharply deteriorates; this likely happens as the dynamics is not properly developed on the RM. In other words, MMALA appears to confine the Langevin dynamics somewhat incorrectly, a feature more visible with an increased quantum of observation data. The slight ambiguity in the performance of HMC may be attributed to suboptimal tuning.

	GALA	**MALA**	**MMALA**	**HMC**
Rayleigh (True $\sigma = 2$)				
burn-in	48,54,68	27,29,48	338,397,500	11,13,23
Acceptance (%)	90.05,90.75, 92.4	76.5, 77.85, 79.95	86.85,88.95, 90.25	77.2, 78.72, 81.1
Estimated mean	2.0298,2.0353, 2.0396	2.031,20.361, 2.0399	2.0206,2.0523, 2.0384	2.0345,2.0390, 2.0440
Sample variance	[1.8,2.1,2.7] $\times 10^{-3}$	[2.9,3.1,3.3] $\times 10^{-3}$	[2.8,4.5,8] $\times 10^{-3}$	[4.6,5.1,6.1] $\times 10^{-3}$
Runtime (seconds)	1.0605	0.9821	1.0904	2.1387
Banana (True $B = 0.1$)				
burn-in	9,11,12	12,13,14	-	54,59,63
Acceptance (%)	100,100, 100	83.7,84.7, 86.7	-	94.3, 95.02, 96.2
Estimated mean	0.1005,0.1005, 0.1005	0.1006,0.1007, 0.1008	-	0.1005,0.1005, 0.1006
Sample variance	[0.8,2,3] $\times 10^{-8}$	[2.4,3.1,4.03] $\times 10^{-6}$	-	[2,2.2,2.5] $\times 10^{-5}$
Runtime (seconds)	0.2	0.25	0.42	0.48

Table 3.1: Comparison of various performance metrics (minimum, median and maximum) for 2000 posterior samples obtained over 10 independent runs of each method. 200 and 10 observations are used for the Rayleigh and Banana distribution, respectively. The mean and sample variance are calculated based on 1000 samples after discarding the burn-in samples for each method.

Banana-shaped distribution. Next, consider the 2-dimensional banana-

shaped distribution, the joint probability density of which is given by

$$p_{xy}(x,y;B) = \exp\left(-\frac{x^2}{200} - \frac{1}{2}(y+Bx^2-100B)^2\right) \tag{3.23}$$

The banana-shaped distribution is basically a twisted Gaussian distribution with a twist parameter B and forms a good test distribution in the context of problem geometry. Our interest, as in the previous example, is in estimating the parameter B. Detailed derivation of the stochastically developed equation for this distribution is given in the Appendix B. Figure 3.3 compares results in the burn-in phase for the twist parameter reconstructed by GALA, MALA and HMC given a set of only 10 sample points. The MMALA method fails for this problem. Similar to the Rayleigh distribution, in this problem too HMC requires integration of Hamiltonian dynamics over 50 time steps for one proposal, considerably increasing the computational cost. Again, Table 3.1 gives a summary of the various performance metrics for the methods considered. GALA performs better compared to other methods for all the metrics considered, particularly in the sample variance which is 4 orders of magnitude lower than HMC and 3 orders lower than MALA whilst taking the least computation time. Indeed, given the fully connected nature of curves in one dimen-

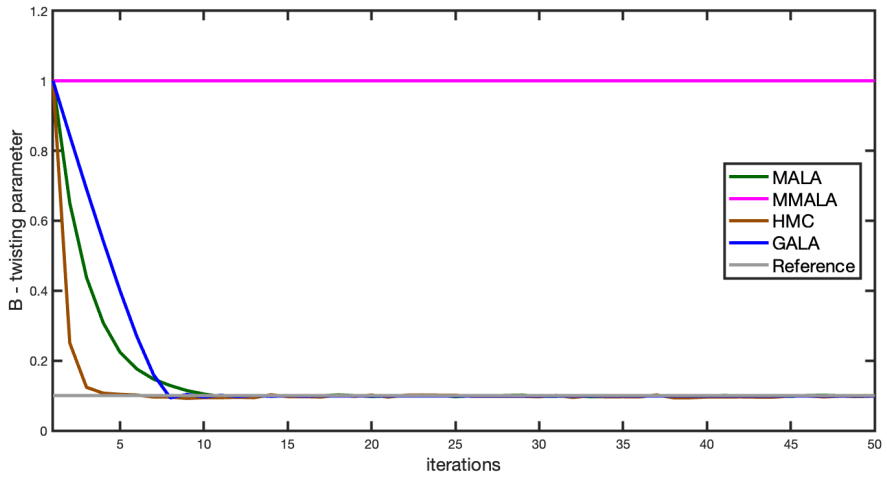
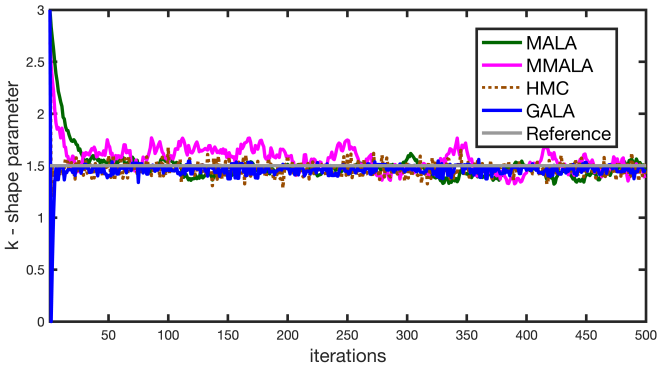


Figure 3.3: Parameter B in the Banana-shaped distribution via GALA ($\Delta t = 0.1$), MALA ($\Delta t = 0.000005$), MMALA ($\Delta t = 0.1$) and HMC ($\Delta t = 0.0001, L = 50$) for $N = 10$ sample observations

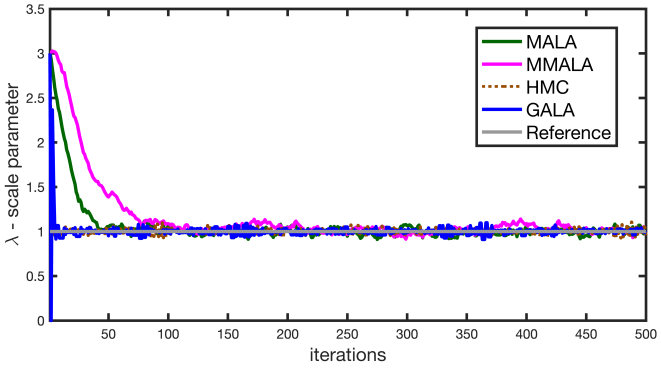
sion, the full potential of a Riemannian geometric method such as GALA is not realized for 1D cases, as in the Rayleigh and banana-shaped distribution problems. In what follows, we consider a few higher dimensional illustrations to showcase the potential benefits of GALA.

Weibull distribution. The Weibull distribution is 1-dimensional and characterised by two parameters. This heavy-tailed distribution is important as it can be used to represent many different shapes by appropriately choosing the two parameters (viz. the shape parameter, k and the scale parameter, λ). The current estimation problem therefore involves these two parameters. The shape of the probability density is very sensitive to changes in the parameter k. A detailed derivation of the developed equation for this distribution is given in the Appendix B. Figure 3.4 gives a comparison of results through GALA, MALA, MMALA and HMC in the burn-in phase. The estimation by GALA, which is manifestly of a superior quality vis-á-vis MALA and MMALA, is only matched by the HMC. Table 2 gives a comparison of the various performance metrics. GALA performs better than other methods overall, particularly the variance, which is at least one order of magnitude lower than the other methods. However, owing to a complex nature of the gradients with respect to the desired parameters, the expectations appearing in the FIM have been numerically evaluated for GALA and MMALA, see Appendix B for the expressions. Note that this issue could either possibly be solved analytically, or accelerated numerically (since numerical expectations can be parallelised), and only appears for very specific distributions. This aspect may be borne in mind whilst assessing the reported comparisons of these two methods with MALA and HMC, particularly the computation time.

Multivariate Gaussian distribution. Now consider a multivariate Gaussian distribution. Again, the detailed derivation for the developed equation is included in the Appendix B. In order to better understand the performance variation of different methods with increasing dimensionality, we consider a sequence of problems with number of parameters to be estimated varying from 5 to 65. Figure 3.5 shows the chain plots for the 65-dimensional parameter problem with 10 unknown mean and



(a) k



(b) λ

Figure 3.4: Parameters of the Weibull distribution via GALA ($\Delta t = 0.1$), MALA($\Delta t = 0.0005$), MMALA ($\Delta t = 0.1$) and HMC ($\Delta t = 0.01, L = 10$) for $N = 400$ sample observations; (a) shape parameter - k; (b) scale parameter - λ

55 covariance matrix components (i.e. for a 10-dimensional Gaussian distributed dataset) obtained by GALA, MALA, MMALA, corrected MMALA [65] and Stan, for a few components of the mean vector and the covariance matrix. All methods except GALA fail for the 65-dimensional problem. For most of the components, MALA does not converge in the 1000 steps considered. MMALA diverges, and all samples after about 400 iterations are rejected. Stan (brms package in R) converges for all the mean components; however, out of the 55 components of the covariance matrix, it only converges to correct values for one or two. For the remaining components, it converges to incorrect parameter values. The computation time with Stan is also (at least) five times more than GALA for this particular problem.

Figure 3.6 gives the variation in performance as well as computation time of several methods with increasing dimension. Specifically, it gives the minimum

	GALA	MALA	MMALA	HMC
Weibull (True $\lambda = 1$, True $k = 1.5$)				
burn-in	8,8,9	24,33,56	137,180,254	4,16,39
Acceptance (%)	95.15,95.8, 96.45	100,100, 100	100,100, 100	98.95, 99.45, 99.7
Estimated mean λ	1.03,1.031, 1.031	1.029,1.031, 1.036	0.999,1.043, 1.075	1.03,1.032, 1.033
Sample variance λ	[5.9,6.6,7.4] $\times 10^{-5}$	[0.98,1.1,1.2] $\times 10^{-3}$	[1.3,2.5,4.3] $\times 10^{-3}$	[0.89,1.1,1.3] $\times 10^{-3}$
Estimated mean k	1.532,1.533, 1.535	1.52,1.53, 1.54	1.458,1.537, 1.565	1.53,1.534, 1.538
Sample variance k	[1.8,2,2.2] $\times 10^{-4}$	[2.2,2.8,4] $\times 10^{-3}$	[3.9,6,9.8] $\times 10^{-3}$	[2.6,3,3.1] $\times 10^{-3}$
Runtime (seconds)	29.13*	8.23	27.45*	5.32

Table 3.2: Comparison of various performance metrics (minimum, median and maximum) for 2000 posterior samples obtained over 10 independent runs of each method. 400 observations are used for estimation. The mean and sample variance are calculated based on 1000 samples after discarding the burn-in samples for each method. The runtime for GALA and MMALA is unreasonably high due to the numerical expectations used for the Riemannian metric and its derivatives based on 2000 Weibull samples generated every iteration.

and maximum norms of the estimated mean parameter vector across 4 independent chains of length 1000 each (with the last 100 samples if there is no convergence, otherwise with all the samples following burn-in). The gradual performance deterioration of most methods, with GALA being the sole exception is a highlight of this figure. For instance, all methods but Stan (which fails to converge for the cross-covariance term σ_{12}) converge to the correct solution for the 5-parameter (2D Gaussian) problem. For the 9-parameter problem (3D Gaussian) case, all methods but corrected MMALA and GALA fail, even as we observe a markedly slower rate of convergence with corrected MMALA. For still higher dimensional cases, all methods except GALA fail (at least for the 1000 steps over which the simulations are presently performed). Figure 3.6 also displays the computational time of all methods (except MMALA due to rejection of all samples after a few steps) according to dimension. Stan stands out as the method whose computational time increases

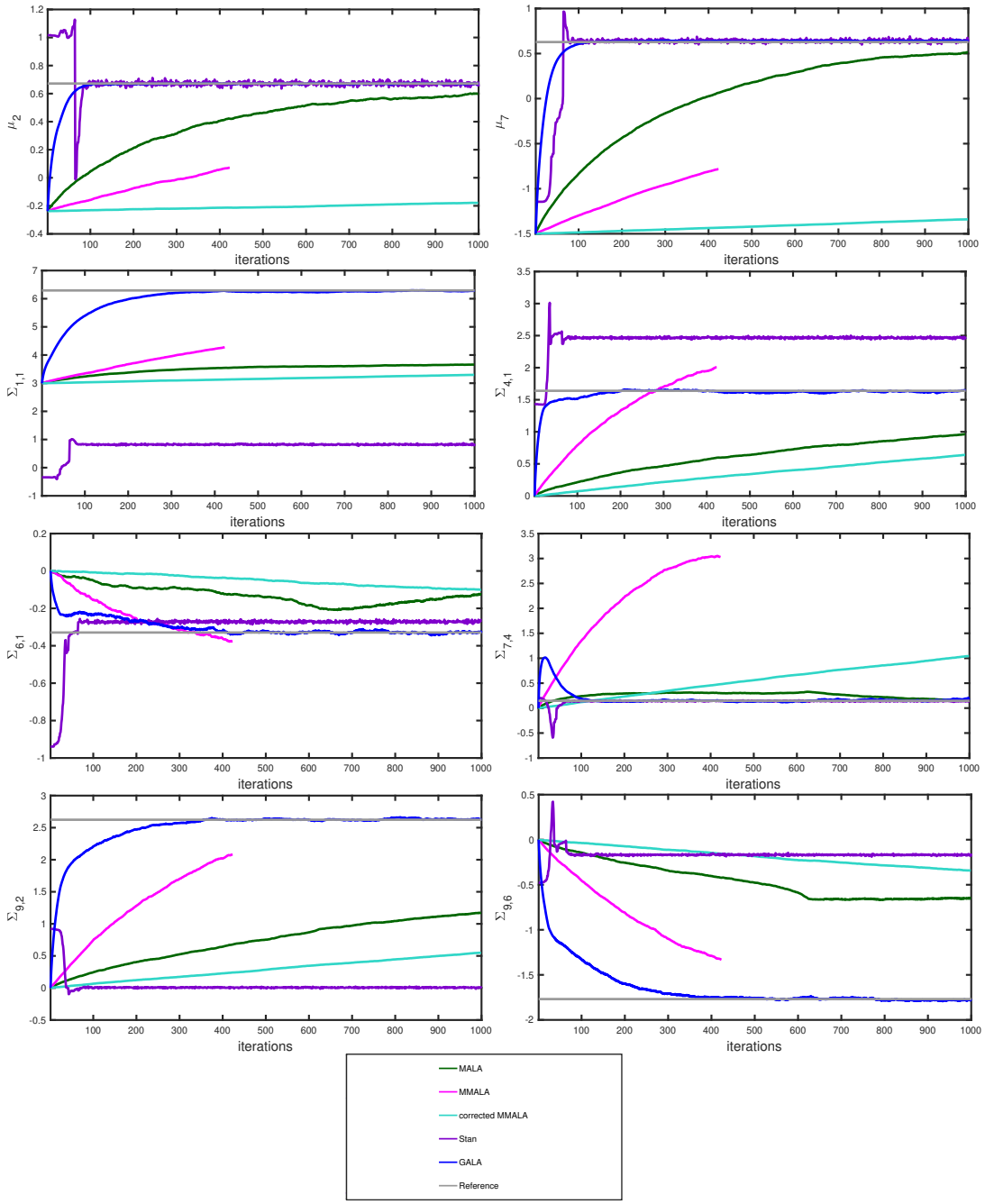


Figure 3.5: A few components of the mean vector and covariance matrix for the 65 parameter multivariate Gaussian distribution via several competing methods; the legend indicating different methods used is shown separately.

the fastest with dimension, whereas MALA, corrected MMALA, and GALA have similar computational times across 5-65 dimensions for this problem.

Figure 3.7 shows the ranges of sample variance and burn-in length for GALA across dimensions 5-65. Again it may be noted that this is only a representative

trend with increasing dimension for only 4 independent chains and may vary with initial conditions. The sample variances do not vary much across dimension, which is an impressive robustness across dimensions. The burn-in lengths increases with the problem dimension, but seemingly linearly. A 5-D problem with a sample size of 1000 requires a burn-in of 50 iterations, whereas a 65-D problem with a sample size of 30000 requires a burn-in of only 400 iterations. These two metrics of performance are demonstrating the unique strength of GALA – as it is the only one to converge towards the solution – and its exceptional efficiency and scalability.

3.5.2 Application to Logistic Regression

In this subsection, we take up a logistic regression problem, one that arises frequently in diverse fields like machine learning, social and medical sciences. Let $X_{D\times N} = \{X^{(1)}, X^{(2)}, X^{(3)} ... X^{(N)}\}$ represent N samples of the D-dimensional explanatory variables that are available along with the binary response variable $t_{1\times N}$. Here each t_i is a Bernoulli random variable with the probability of success depending on $X^{(i)}$. Assuming that the true regression coefficients are represented by $\beta_{(D+1)\times 1}$, the probability of success for each $X^{(i)}_{D\times 1}$ is given by

$$p\left(X^{(i)}|\beta\right) = p\left(t_i = 1\right) = \frac{1}{1+\exp\left(-\beta_0 + \sum_{j=1}^{D}\beta_j X_j^{(i)}\right)}$$

The likelihood of the data is then the product of likelihoods over the N data points. Assuming a prior density on β as $\mathcal{N}(0, \alpha I)$, where α is chosen appropriately, the FIM and its derivative for the posterior are given by

$$G_{pq} = \sum_{i=1}^{N} \frac{\exp\left(-\beta_0 + \sum_{j=1}^{D}\beta_j X_j^{(i)}\right) X_p^{(i)} X_q^{(i)}}{1+\exp\left(-\beta_0 + \sum_{j=1}^{D}\beta_j X_j^{(i)}\right)^2} + \alpha^{-1}\delta_{pq} \tag{3.24}$$

$$\frac{\partial G_{pq}}{\partial \beta_r} = -\sum_{i=1}^{N} \frac{\exp\left(-\beta^T \bar{X}^{(i)}\right) \bar{X}_p^{(i)} \bar{X}_q^{(i)} X_r^{(i)}}{\left(1+\exp\left(-\beta^T \bar{X}^{i}\right)\right)^2} + 2\sum_{i=1}^{N} \frac{\left(\exp\left(-\beta^T \bar{X}^{(i)}\right)\right)^2 \bar{X}_p^{(i)} \bar{X}_q^{(i)} \bar{X}_r^{(i)}}{\left(1+\exp\left(-\beta^T \bar{X}^{(i)}\right)\right)^3} \tag{3.25}$$

See the Appendix B for a detailed derivation. Thus, all the input quantities

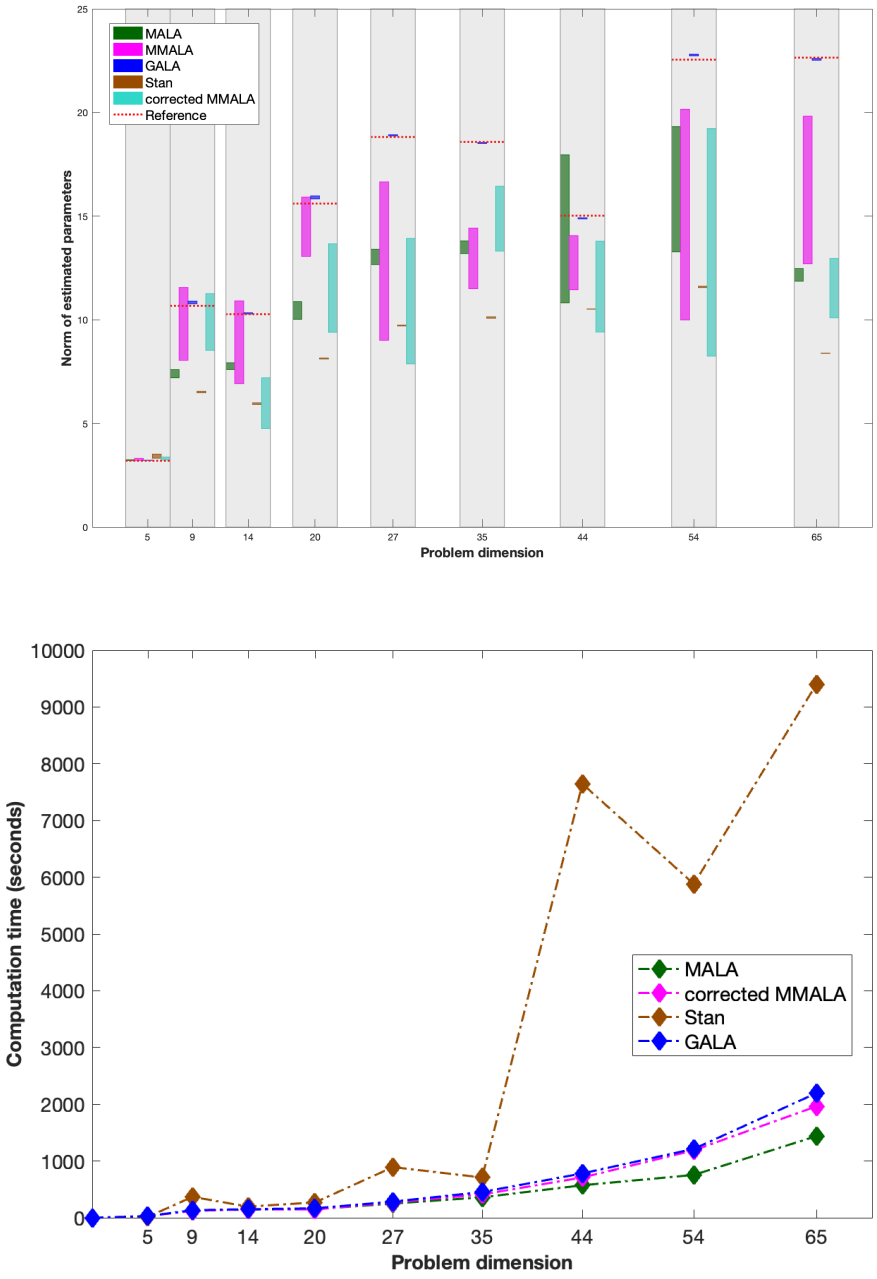


Figure 3.6: Top: Minimum and maximum of estimated parameter norms across 4 independent chains of length 1000 each for Gaussian problems with varying dimensions. The means are based on the samples after burn-in if convergence occurs, otherwise it is determined using the last 100 samples. Bottom: A comparison of computation time. MMALA is not included since after a few steps, all samples are typically rejected, which renders a comparison inappropriate.

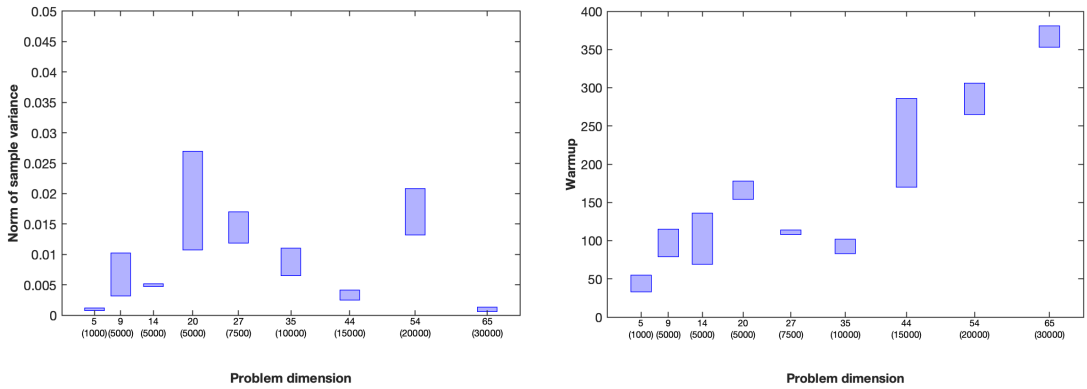


Figure 3.7: Left: ranges of sample variance after burn-in for GALA across the 4 independent chains for Gaussian problems with varying dimensions. Right: ranges of burn-in for GALA for across 4 independent chains. The bracketed labels on $X-$ axis indicate the size of dataset used.

needed for GALA are now determined. We show in Figure 3.8 a comparison of results via GALA, MALA, MMALA and Stan (rstanarm package in R) for a few regression parameters. For the 30-dimensional problem considered, the parameters are chosen so that 25 of them are uniformly distributed in $[0, 15]$ while the remaining 5 are uniformly distributed in $[-15, -10]$. This is done to make the problem slightly more challenging. MALA just about converges in the 3000 steps for this problem. MMALA is faster than MALA though much slower than GALA; whereas Stan, even though it is the fastest, fails for this problem. The burn-in with Stan are automatically discarded, which is reflected in the figures. Again, Table 3 summarizes the various performance metrics. Similar to the Gaussian example, in this problem too, the norm of the 30-dimensional mean and variance is given for convenience.

3.6 Concluding remarks

Exploiting the FIM as a Riemannian metric and the associated Riemannian connection, we have stochastically developed a given SDE in the standard Euclidean setting. Unlike the known equation for a Brownian motion developed on the RM, the SDE that we geometrically adapt has a non-trivial drift term as well. We have specifically used this novel construction to modify the Langevin SDE and hence MALA. Our anticipation had been that a restriction of solutions to the Riemannian hypersurface should yield significantly higher accuracy and faster convergence,

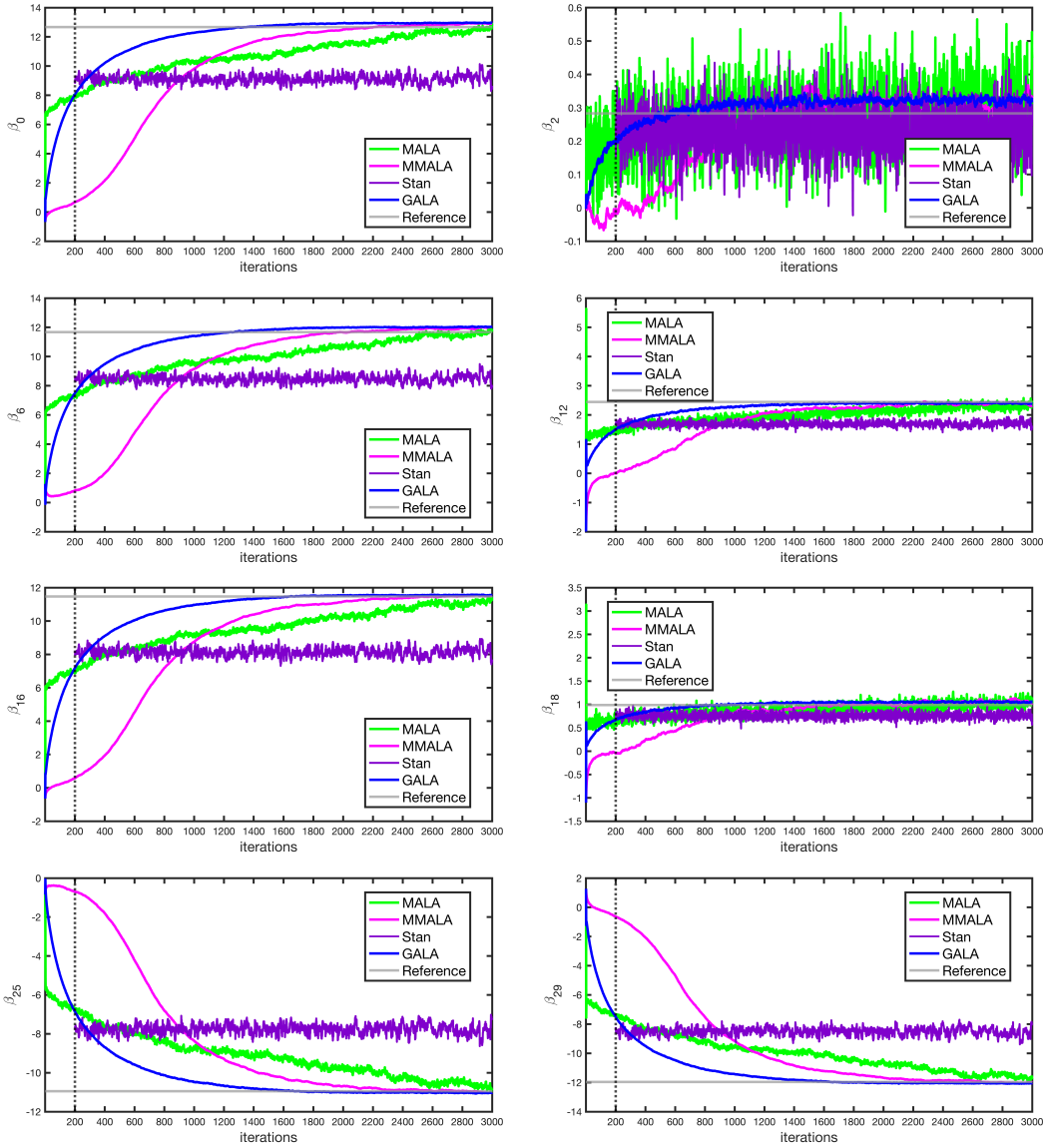


Figure 3.8: A few of the reconstructed regression parameters for a 30 dimensional logistic regression problem via several methods.

	GALA	MALA	MMALA	Stan
Logistic Regression (30 dimensional), norm of true mean $= 48.62$				
burn-in	1348,1363, 1294,1336	-	2222,2203, 2112,2185	-
Acceptance (%)	100,100,100,100	-	100,100,100,100	-
Norm of estimated mean	48.81,48.8, 48.82,48.92	-	48.9,48.85, 48.82,48.76	-
Norm of sample variance	0.01,0.0091, 0.01,0.02	-	0.016,0.01, 0.02,0.01	-
Runtime (seconds)	966	819	852	≈ 160

Table 3.3: Comparison of various performance metrics for 3000 samples obtained for the logistic regression problem for each of the 4 independent runs of each method. Whereas for the logistic regression problem, the estimates of the norm of mean and sample variance are based on 500 samples after discarding burn-in. The logistic regression codes for GALA and MMALA are parallelised to achieve a 40 % reduction in computation time, while that for MALA is not.

even though Brownian noise processes have unbounded variations. That this feature can indeed be realized is demonstrated through a couple of applications, e.g. estimating the parameters in a probability density given a set of observations and solving the logistic regression problem. For both problems, the GALA based approach far outperforms the standard MALA and HMC, both in faster burn-in and estimation accuracy alike (e.g. sample variance smaller by orders of magnitude). This relative superiority of performance is generally more pronounced as the problem dimension increases, and so is the superiority in computational cost compared to Stan. This is particularly noticeable in the estimation of covariance in multivariate normal distributions, where it is the only successful method.

Beyond performance, and scalability to high dimensions, we also want to highlight the accessibility of GALA compared to HMC and NUTS (Stan). Indeed, HMC requires tuning of two parameters for it to work efficiently, which becomes difficult with an increase in dimension. The NUTS sampler was developed with the objective to get around this very difficulty, and claimed to perform at least as well as HMC. However, we did not find Stan to be accessible. The installation on a Linux or Windows machine for the MATLAB or R implementation of Stan (software to

implement the NUTS sampler) failed, despite multiple attempts and several hours of professional help from research software engineers. We finally moved to R on Mac OS to successfully implement the 'rstan' package. Even so, except for a few standard distributions, just to set up the problem requires a fair bit of knowledge to write a program in Stan. To overcome this, we worked with 'rstanarm' for the logistic regression problem and 'brms' package in R for the multivariate Gaussian problem. A few packages including the 'rstanarm' and 'brms' [76, 77] were developed to bypass the need for a user to program in Stan, which is useful, but one package may be more straightforward than another for a given problem. Leaving aside the difficulties in installation and the posing of the problem in Stan, we observe from the results, that it converges to incorrect values for both large dimensional examples considered in this work, viz. the 30-dimensional logistic regression problem and the 65-dimensional Gaussian parameter estimation problem. Indeed, this issue was explored in [78], and it was found that NUTS does not converge to the correct invariant distribution, although it could be achieved with some modification. In contrast, GALA is easy to implement, only a reasonable choice of dt allows the algorithm to function efficiently, the value of dt we have used is typically one or two orders of magnitude higher than that used for MALA. Unlike all first gradient based methods, GALA requires the derivative of the FIM, which we provide in the Appendix B for all the examples considered in this work, for a ready reference.

A word about a possible future direction before concluding this chapter. The continuous but non-differentiable structure of the B.M. requires that we write the SDEs in terms of differentials and not the usual derivatives. The derivation of the developed equations on an RM, as in Section A.1, therefore required the language of exterior calculus and Cartan's structure equations. Although the curvature tensor, or more precisely the curvature 2-form which is a fundamental tensor field of incompatibility on an RM, has appeared in our developed equation, we have presently neglected it as a higher order term. An understanding and exploitation of this term in the context of Monte Carlo algorithms is, to our understanding, an important element of future study, in particular while working in high dimensional spaces. A

related curiosity also lies in a possible extension of the geometric framework to a Riemann-Cartan manifold, which would enable the developed dynamics to be further enriched by the torsion 2-form. Overall, the mathematical machinery of Cartan's moving frame appears to be a powerful tool for an insightful understanding of the role of geometry for stochastic development, possibly opening up routes to more efficient Monte Carlo algorithms.

The next chapter is on stochastic optimal control (SOC). It may seem unrelated to the general theme followed so far, and it perhaps turned out to be so, but we started out with an aim to use the stochastically developed SDE with the objective of solving for optimal control. Had this route been feasible, it may have led to the solution of an optimal control problem by only integrating one SDE. However, the construction of the Riemannian metric required derivatives of complicated integrals. We instead modified our approach to avoid this scenario and developed a methodology to solve the optimal control problem that required integrating two Euclidean SDEs. We discuss this approach in detail in the next chapter and how this is different from the existing approaches for SOC using SDEs.

Chapter 4

Time recursive control of stochastic dynamical systems using forward dynamics and applications

This chapter is reproduced in its entirety from our work published in [79]. The only addition here is the concluding paragraph.

4.1 Introduction

A stochastic optimal control (SOC) problem consists in the study of controlling the response of a given dynamical system in the presence of noise such that an associated cost functional is minimized. Problems of this genre arise in several fields of natural science and engineering, a few of the most commonly encountered being finance, robotics, control of vehicles as well as controlling movements in biological systems. If the system is deterministic, the control problem may be solved either by the Hamilton-Jacobi-Bellman (HJB) equation or by the well-known Pontryagin's minimum principle [5, 80]. The latter consists of a system of ordinary differential equations (ODEs) similar to Hamilton's equation of motion but with a mixed boundary condition. Even when the system being considered is stochastic, it may be shown that solving an SOC problem reduces to solving the HJB equation; this is a partial differential equation (PDE) arrived at through the same conceptual underpinnings as in the deterministic case [81]. The underlying basis for the deriva-

tion and solution of the HJB equation for both deterministic and stochastic cases is the dynamic programming principle, a powerful scheme for solving optimization problems by a recursive definition of a value function that pertains to the optimal strategy from the current to the terminal time [5]. The only difference in the deterministic and stochastic cases then is the use of Taylor expansion (in the former) and Ito-Taylor expansion (in the latter) of a certain value function (defined in the next section). This value function comes up naturally while using the principle of dynamic programming. Moreover, for a small class of SOC problems, when the system dynamics and cost functional are of a certain form, the optimal control may in fact be determined by solving ODEs instead of the HJB equation; specifically a Ricatti type equation in the Hessian of the value function.

The discussion so far has been about the solution of continuous-time and continuous state-space control problems. For discrete-time problems, the analog of the HJB equation is the Bellman equation. Problems where the state-space is also discrete and the model dynamics is not explicitly known, fall in the domain of reinforcement learning (RL), which is one of the three major fields of machine learning, the others being supervised and unsupervised machine learning. RL may be looked upon as a way to solve the most general optimal control problems, wherein neither model information nor training data is provided; most common examples being autonomous driving and playing board-games like Go [82]. All that is provided to the RL model is a penalty or reward for its action (in the RL literature, the control is referred to as action), and the objective is to maximize the reward. Interestingly, [83] establishes the connection of stochastic optimal control with RL in continuous space and time. In the present work, we focus on those continuous space and time SOC problems wherein no restriction is put on the process dynamics and cost contribution from the system states; but the control is assumed to act linearly on the dynamics and its contribution to cost is assumed to be quadratic.

The existing methods to solve an SOC problem may be split into two broad categories. The first is a spatio-temporal discretization based approach often using the finite element method; see [84, 85]. A method that, while not quite within

the domain of finite-elements, is nonetheless closely related is studied in [86]. It is a pseudo-spectral method to solve the HJB equation. The discretization based approaches have a major disadvantage in that they are not readily scalable, i.e. it may become computationally demanding, or even intractable, very quickly as the dimension of the system increases. This problem is commonly known in the literature as the curse of dimensionality. The second class of methods is based on Monte Carlo simulations which exploit the Feynman-Kac [6] formula or the so-called non-linear Feynaman-Kac formula [87]. With an appropriate log transformation under certain constraints, the HJB equation may be shown to reduce to a linear PDE of the backward Chapman-Kolmogorov type. This makes it possible to develop schemes to solve the PDE using the Feynman-Kac formula which basically requires simulating SDEs and certain path integrals of solutions of the SDEs, followed by taking appropriate expectations. Although linearization is very helpful from an algorithmic point of view, solutions thus obtained are restricted due to the constraints that are required to enable linearization in the first place. [88, 89, 90] propose efficient schemes based on the linearization of the HJB equation. The non-linear Feynman-Kac, on the other hand, requires simulating a pair of partly coupled SDEs, i.e. the forward-backward stochastic differential equations (FBSDEs); see [91, 92] for an introduction to backward SDEs. [92, 93, 94, 95, 96] propose effective schemes based on backward SDEs. The Monte Carlo route addresses the issue of scalability (which is a major disadvantage of discretization based methods) to a great extent; however it may still be computationally intensive, particularly the schemes based on the non-linear Feynman-Kac. The main reason for this, which will be made clearer in the subsequent sections, is that in order to determine the optimal control at any time, an ensemble of trajectories of the system dynamics must evolve all the way forward till the terminal time and thereafter the backward SDE must evolve all the way back to the current time. Also, the ensemble size may have to be increased significantly as the problem dimension increases. Since the FBSDEs involve integration of SDEs, the ensemble size may also have to be increased appropriately as the time horizon (and/or problem non-linearity) increases, since integrating an SDE

for a longer time results in an increase in the sampling variance with time due to the diffusion term. Recall that sampling variance is the variance of the sample means across multiple simulations, where each simulation has its own sample mean and sample variance.

It is worth noting that while an SOC may fall in different classes such as $\mathscr{L}^1$, $\mathscr{L}^2$ (i.e. the cost associated with the control force appears as an $\mathscr{L}^1$ or $\mathscr{L}^2$ norm), finite time horizon, with a stopping time etc., if a particular class of problems is solved by a certain method, it may not be too difficult to adapt it to the others. Presently, we work with the most commonly encountered class of optimal control problems i.e. the linear-quadratic (LQ) problem over a finite time horizon (defined more precisely later). As will be seen, while we do not consider the different classes, the proposed scheme should admit a ready adaptation to a cost functional with a stopping time and to problems in $\mathscr{L}^1$ control. The same is the case with problems over an infinite time horizon. Also, we do not discuss the existence and uniqueness of solutions of an optimal control problem which may be considered a mathematical field of study in its own right; see [97, 98, 99] for an introduction. We assume throughout that the necessary and sufficient conditions for a solution to exist are satisfied. In this context, we may note that the notion of a viscosity solution to the HJB equation may be exploited in order to considerably relax the continuity requirements of the fields involved [100].

In recent years, the control of mechanical oscillators has received considerable attention. Interesting applications of SOC for system dynamics pertaining to that of mechanical oscillators are explored in several articles. In [101, 102], for instance, the control is designed for energy harvesting in order to accomplish maximum power generation. Control of mechanical oscillators where the control force is bounded is investigated in [103, 104]. In [105], a proposal to control a chaotic Duffing-Holmes oscillator is put forth through an appropriate choice of the external periodic excitation. A probabilistic approach to determine the control of oscillators is discussed in [106, 107]. Oscillator based CPG (central pattern generator) models are also at the core of the theory for controlling joints in robots [108]. In a semi-

nal article [109], it was shown that identical chaotic systems may be synchronized via an appropriate coupling. This synchronized coupling of chaotic systems finds application in the field of communications and signal processing [110]. An adaptive control method applied to the synchronization of two oscillators is discussed in [111]. Synchronization and control of chaos in non-linear oscillators are studied in fair detail in [112]. Wiercigroch and co-workers have also contributed substantively to the control of smooth and non-smooth mechanical oscillators in the context of various applications such as sea wave energy extraction; see [113, 114, 115]. Incidentally, the importance of the subject of SOC may also be gauged from the fact that it is intimately connected to non-equilibrium stochastic thermodynamics of small systems; see [116]. An SOC based interpretation could, in particular, be beneficial in a more nuanced representation of the second law of thermodynamics, or of Jarzynski's equality [117].

Another class of systems whose control could prove quite beneficial, even though it may appear counter-intuitive, is a chaotic system. Two major articles that first explored this class of problems are [118, 119]. The basic property of a chaotic system is that small, even infinitesimal, changes in the initial states lead to large changes later on. This property, known as the 'butterfly effect', is what makes the system dynamics chaotic. This seemingly problematic property is exactly what could be exploited, at least in principle, to achieve major changes in the system behaviour by applying small controls at appropriate times, thereby reducing the total cost of control. One way to minimize or reduce the cost is to delay the application of control until the system has reached a point near the desired orbit; this is only possible in chaotic systems since there are infinitely many unstable periodic orbits within the strange attractor and the system is bound to reach the desired orbit (or one nearby) at some point in time. If the application of control is delayed until this happens, it is possible to reduce the cost dramatically, since thereafter a relatively small control force might be needed in order to maintain the system in the desired orbit. A remarkable example that illustrates the importance of controlling a chaotic system is discussed in [118] – the spacecraft ISEE-3/ICE could be sent more than

50 million miles across the Solar system to bring about the first cometary encounter only because the nature of the underlying dynamics was chaotic.

In this work, we propose a numerical scheme based on Monte Carlo simulations for solving an SOC problem. We replace the terminal cost by its Ito expansion about an initial time; this enables the derivation of a PDE which is subject to an initial condition unlike the HJB equation which is subject to a terminal condition. Therefore, it is possible to write a pair of SDEs corresponding to the proposed PDE such that both of them need to be integrated forward in time. In some sense, which will be explicated in the subsequent sections, the proposed scheme may be considered complementary to the FBSDE method of solving an SOC problem. While the total cost being minimized remains the same and the same control objectives are fulfilled, pathwise solutions thus obtained may not be the same as the ones via the HJB equation; this however largely depends on the system dynamics. The resulting numerical scheme presently derived is essentially the simulation of a pair SDEs plus a curve fitting at every time step. Therefore it has significant benefits in terms of computational expedience and scalability over the FBSDE method. It is also much simpler to implement. The scheme exhibits lower sampling variance and robustness to non-linearity in the system dynamics. We illustrate the advantages of this approach via several numerical examples, mainly involving non-linear mechanical oscillators in their chaotic as well as non-chaotic regimes.

The rest of this chapter is organized as follows. Section 4.2 provides a brief review of the SOC, including a derivation of the HJB equation as well as its linearized version and an outline of an existing scheme based on FBSDEs. Section 4.3 gives a detailed account of the proposed method, wherein two alternative derivations are provided to arrive at the same PDE subject to an appropriate initial condition. The derivation is also specialized for the case of mechanical oscillators since their dynamics in the state space are not straightforward SDEs; the associated pair of SDEs is arrived at using Ito's formula and finally a pseudo-code is included for clarity. Next, Section 4.4 illustrates the proposed method with several examples. A simple 1-dimensional problem is first included for a comparison with the theoretical solu-

tion as well as the basic FBSDE method [93], followed by a 2D Ornstein-Uhlenbeck process, the Lorenz 1963 oscillator and finally two nonlinear mechanical oscillators, viz. the hardening Duffing and the Duffing-Holmes. Throughout this chapter, we use lowercase Roman or Greek letters to denote scalars or scalar valued functions or realizations of scalar-valued random variables. Scalar-valued random variables are denoted by uppercase letters. Bold lowercase letters are used for vectors, vector-valued functions or realizations of vector-valued random variables. We use bold uppercase letters for vector-valued random variables (or stochastic processes, as appropriate) and uppercase non-italicized (text) letters for matrices or matrix-valued functions. The distinction between a constant scalar/vector/matrix and a scalar/vector/matrix valued function is made clear by writing out the dependencies. For instance, c is a scalar, $c(\boldsymbol{x})$ is a scalar valued function of the vector $\boldsymbol{x}$, $\boldsymbol{X}_t := \boldsymbol{X}(t)$ is a vector-valued stochastic process (whose realization is $\boldsymbol{x}_t$) and C is a matrix. The path of a scalar field $u_t := u(t)$ over a time interval $[t_1, t_2]$ is represented as $u_{t_1 \to t_2}$. Whenever there is an ambiguity about the time at which a state is being referred, it is given with a subscript, i.e. $\boldsymbol{x}_t$ instead of $\boldsymbol{x}$.

4.2 A brief review of stochastic optimal control

For clarity and completeness of the exposition, we give a brief review of stochastic optimal control for the LQ case, starting with the problem definition. This is followed by a brief derivation of the associated HJB equation, a PDE in the extremal cost. This should be helpful in contrasting the FBSDE approach to solve the HJB equation and a purely forward SDE based approach as proposed in the next section. We also discuss an existing numerical scheme based on FBSDEs. For the existence and uniqueness of solutions to this class of control problems, we refer to [97].

Let $(\boldsymbol{\Omega}, \mathscr{F}, \mathscr{P})$ denote a complete probability space supplied with the natural filtration $\{\mathscr{F}_t\}$. Here $\boldsymbol{\Omega}$ is the sample space, $\mathscr{F}$ the sigma algebra and $\mathscr{P}$ the probability measure. Note that, for completeness, $\mathscr{F}$ contains all the $\mathscr{P}$-null events. Consider a system whose dynamics is represented by the following SDE

$$d\boldsymbol{X}_t = \boldsymbol{\alpha}(t, \boldsymbol{X}_t)dt + \Sigma(t, \boldsymbol{X}_t)d\boldsymbol{B}_t \tag{4.1}$$

where $\boldsymbol{X}_t : \boldsymbol{\Omega} \times \mathbb{R}^+ \to \mathbb{R}^d$ is a vector-valued stochastic process, $\boldsymbol{\alpha} : \mathbb{R}^+ \times \mathbb{R}^d \to \mathbb{R}^d$ is the drift vector field, $\Sigma : \mathbb{R}^+ \times \mathbb{R}^d \to \mathbb{R}^{d\times d}$ is the diffusion matrix field (presently assumed to be square) and $\boldsymbol{B}_t \in \mathbb{R}^d$ is a vector Brownian motion with independently evolving standard scalar components. We assume the drift and the diffusion fields above to be Lipschitz continuous and with a sublinear growth in the first argument to ensure existence and uniqueness of solutions to the SDE. In this work, Σ is taken as constant; hence in what follows $\Sigma(t,\boldsymbol{X}_t)$ is replaced with Σ. Let the linearly controlled dynamics pertaining to (4.1) be represented as

$$d\boldsymbol{X}_t = \boldsymbol{\alpha}(t,\boldsymbol{X}_t)dt + \mathrm{G}(t,\boldsymbol{X}_t)\boldsymbol{u}_t dt + \Sigma(t,\boldsymbol{X}_t)d\boldsymbol{B}_t \tag{4.2}$$

where $\mathrm{G} : \mathbb{R}^+ \times \mathbb{R}^d \to \mathbb{R}^{d\times d}$ is a square matrix appearing with $\boldsymbol{u}_t \in \mathbb{R}^d$, the control vector. The objective of the control problem is to find an optimal control trajectory $\boldsymbol{u}_{t_0\to t_f} = \boldsymbol{u}^*_{t_0\to t_f}$ such that, starting with $\boldsymbol{x}_0 := \boldsymbol{x}(t_0)$, the following cost functional which is quadratic in $\boldsymbol{u}$ is minimized.

$$c(t_0,\boldsymbol{x}_0,\boldsymbol{u}_{t_0\to t_f}) = \mathbb{E}\left[g(\boldsymbol{X}_{t_f}) + \int_{t_0}^{t_f}(q(t,\boldsymbol{X}_t) + p(\boldsymbol{u}_t))dt\right] \tag{4.3}$$

Here $q(t,\boldsymbol{X}_t)$ is the running cost on $\boldsymbol{X}_t$, $p(\boldsymbol{u}_t)$ is the running cost on $\boldsymbol{u}_t$ which is taken as quadratic for this work, i.e. $p(\boldsymbol{u}_t) = \frac{1}{2}\boldsymbol{u}_t^T \mathrm{R}\boldsymbol{u}_t$ where $\mathrm{R} \in \mathbb{R}^{m\times m}$ is a known constant matrix. The form of $p(\boldsymbol{u}_t)$ shows that the total running cost involving the control term is interpreted as an $\mathscr{L}^2$ integral. To proceed further, one typically defines the value function as the minimal cost to reach some final state $\boldsymbol{x}_{t_f}$, given that the starting point at time t is fixed at $\boldsymbol{x}$, as follows.

$$v(t,\boldsymbol{x}) = \min_{\boldsymbol{u}_{t\to t_f}} c(t,\boldsymbol{x},\boldsymbol{u}_{t_0\to t_f}) \tag{4.4}$$

A recursion in v – which proves useful in finally writing a PDE in v – may be arrived

at as follows; see [90] for instance.

$$
\begin{aligned}
& v(t,\boldsymbol{x}) \\
&= \min_{\boldsymbol{u}_{t\to t_f}} \mathbb{E}\left[g(\boldsymbol{X}_{t_f}) + \int_t^{t_f} (q(s,\boldsymbol{X}_s) + p(\boldsymbol{u}_s))ds \middle| \boldsymbol{X}_t = \boldsymbol{x} \right] \\
&= \min_{\boldsymbol{u}_{t\to t_f}} \mathbb{E}\left[g(\boldsymbol{X}_{t_f}) + \int_t^{t+dt} (q(s,\boldsymbol{X}_s) + p(\boldsymbol{u}_s))ds + \int_{t+dt}^{t_f} (q(s,\boldsymbol{X}_s) + p(\boldsymbol{u}_s))ds \right] \\
&= \min_{\boldsymbol{u}_{t\to t+dt}} \mathbb{E}\Bigg[\left(\min_{\boldsymbol{u}_{t+dt\to t_f}} \mathbb{E}\big[g(\boldsymbol{X}_{t_f}) + \int_{t+dt}^{t_f} (q(s,\boldsymbol{X}_s) + p(\boldsymbol{u}_s))ds \big| \boldsymbol{X}_{t+dt} = \boldsymbol{x}_{t+dt} \big] \right) \\
&\qquad\qquad + (q(t,\boldsymbol{x}) + p(\boldsymbol{u}_t))dt \Bigg] ; \text{for any } \boldsymbol{x}_{t+dt} \\
&= \min_{\boldsymbol{u}_{t\to t+dt}} \mathbb{E}\Big[v(t+dt,\boldsymbol{X}_{t+dt}) + \big(q(t,\boldsymbol{x}) + p(\boldsymbol{u}_t)\big)dt \Big] \qquad (4.5)
\end{aligned}
$$

By way of an interesting digression, we may add that an SOC problem also admits representation as a martingale problem. Specifically, only if a realized path $\boldsymbol{x}_t$ is obtained with the optimal control $\boldsymbol{u}^*$, then the following stochastic process is a martingale.

$$
M_t^{\boldsymbol{u}^*_{t_0\to t_f}} = v(t,\boldsymbol{x}) + \int_{t_0}^{t_f} (q(s,\boldsymbol{X}_s) + p(\boldsymbol{u}_s^*))ds \qquad (4.6)
$$

Indeed, for $\boldsymbol{u}_{t_0\to t_f} \neq \boldsymbol{u}^*_{t_0\to t_f}$, the process $M^{\boldsymbol{u}}_{t_0\to t_f}$ is a submartingale; see [120]. Now, the Ito-Taylor expansion [26] of $v(t+dt,\boldsymbol{x}_{t+dt})$ about $(t,\boldsymbol{x}_t)$ gives

$$
\begin{aligned}
\mathbb{E}[v(t+dt,\boldsymbol{X}_{t+dt})] = \mathbb{E}\Bigg[& v(t,\boldsymbol{x}_t) + \frac{\partial v(t,\boldsymbol{x}_t)}{\partial t}dt + \left(\frac{\partial v(t,\boldsymbol{x}_t)}{\partial \boldsymbol{x}}\right)^T (\boldsymbol{\alpha}(t,\boldsymbol{x}_t) + \mathrm{G}\boldsymbol{u})dt \\
& + \frac{1}{2}\left(\frac{\partial^2 v(t,\boldsymbol{x}_t)}{\partial \boldsymbol{x}^2} : \Sigma\Sigma^T \right)dt + \left(\frac{\partial v(t,\boldsymbol{x}_t)}{\partial \boldsymbol{x}}\right)^T \Sigma d\boldsymbol{B}_t \Bigg] \qquad (4.7)
\end{aligned}
$$

where ^{T} denotes transposition and : Frobenius inner product of matrices, i.e. $\mathrm{C} : \mathrm{D} = \sum_{i,j} \mathrm{C}_{ij}\mathrm{D}_{ij}$. Since $v(t,\boldsymbol{x}_t)$ is fixed and the expectation of the last term on the right hand side above is zero, the last equation becomes

$$
\begin{aligned}
\mathbb{E}[v(t+dt,\boldsymbol{X}_{t+dt})] = v(t,\boldsymbol{x}_t) & + \frac{\partial v(t,\boldsymbol{x}_t)}{\partial t}dt + \left(\frac{\partial v(t,\boldsymbol{x}_t)}{\partial \boldsymbol{x}}\right)^T (\boldsymbol{\alpha}(t,\boldsymbol{x}_t) + \mathrm{G}\boldsymbol{u})dt \\
& + \frac{1}{2}\left(\frac{\partial^2 v(t,\boldsymbol{x}_t)}{\partial \boldsymbol{x}^2} : \Sigma\Sigma^T \right)dt \qquad (4.8)
\end{aligned}
$$

Substitution of this in equation (4.5) leads to

$$v(t,\boldsymbol{x}) = \min_{\boldsymbol{u}_{t\to t+dt}} \Bigg[v(t,\boldsymbol{x}) + \frac{\partial v(t,\boldsymbol{x})}{\partial t}dt + \left(\frac{\partial v(t,\boldsymbol{x})}{\partial \boldsymbol{x}}\right)^T (\boldsymbol{\alpha}(t,\boldsymbol{x}) + \mathrm{G}\boldsymbol{u})dt + \frac{1}{2}\left(\frac{\partial^2 v(t,\boldsymbol{x})}{\partial \boldsymbol{x}^2} : \Sigma\Sigma^T\right)dt + \big(q(t,\boldsymbol{x}) + p(\boldsymbol{u})\big)dt \Bigg] \quad (4.9)$$

$$-\frac{\partial v(t,\boldsymbol{x})}{\partial t} = \min_{\boldsymbol{u}_{t\to t+dt}} \Bigg[\left(\frac{\partial v(t,\boldsymbol{x})}{\partial \boldsymbol{x}}\right)^T (\boldsymbol{\alpha}(t,\boldsymbol{x}) + \mathrm{G}\boldsymbol{u}) + \frac{1}{2}\left(\frac{\partial^2 v(t,\boldsymbol{x})}{\partial \boldsymbol{x}^2} : \Sigma\Sigma^T\right) + q(t,\boldsymbol{x}) + p(\boldsymbol{u}) \Bigg] \quad (4.10)$$

Since the function on the right hand side is quadratic in $\boldsymbol{u}$, the optimal control strategy can be found in closed form. It may be obtained by setting the derivative of the right hand side of equation 4.10 with respect to $\boldsymbol{u}$ to zero.

$$\frac{\partial}{\partial \boldsymbol{u}}\left[\left(\frac{\partial v(t,\boldsymbol{x})}{\partial \boldsymbol{x}}\right)^T \mathrm{G}\boldsymbol{u} + \frac{1}{2}\boldsymbol{u}^T \mathrm{R}\boldsymbol{u}\right] = 0 \quad (4.11)$$

Solving the equation above, the optimal control as a field may be obtained as follows.

$$\boldsymbol{u}^*(t,\boldsymbol{x}) = -\mathrm{R}^{-1}\mathrm{G}^T \frac{\partial v(t,\boldsymbol{x})}{\partial \boldsymbol{x}} \quad (4.12)$$

Finally, the HJB equation may be arrived at upon substituting the optimal control strategy $\boldsymbol{u}^*(t,\boldsymbol{x})$ in equation 4.10.

$$\frac{\partial v(t,\boldsymbol{x})}{\partial t} + \left(\frac{\partial v(t,\boldsymbol{x})}{\partial \boldsymbol{x}}\right)^T \boldsymbol{\alpha}(t,\boldsymbol{x}) + \frac{1}{2}\left(\frac{\partial^2 v(t,\boldsymbol{x})}{\partial \boldsymbol{x}^2} : \Sigma\Sigma^T\right) + q(t,\boldsymbol{x}) - \frac{1}{2}\left(\frac{\partial v(t,\boldsymbol{x})}{\partial \boldsymbol{x}}\right)^T \mathrm{G}\mathrm{R}^{-1}\mathrm{G}^T \frac{\partial v(t,\boldsymbol{x})}{\partial \boldsymbol{x}} = 0 \text{ ; subject to } v(t_f,\boldsymbol{x}) = g(\boldsymbol{x}) \quad (4.13)$$

Thus, in order to determine the optimal control for (4.2), the HJB equation (4.13) must be solved. This problem may be approached in two ways. The first is to linearize the HJB equation by the following log transformation.

$$v(t,\boldsymbol{x}) = -\tau \log(\phi(t,\boldsymbol{x})) \quad (4.14)$$

The result is the following PDE in $\phi(t,\boldsymbol{x})$, subject to $\phi(t_f,\boldsymbol{x}) = \exp\left(-\frac{1}{\tau}g(\boldsymbol{x})\right)$.

$$\frac{\partial\phi(t,\boldsymbol{x})}{\partial t} + \left(\frac{\partial\phi(t,\boldsymbol{x})t}{\partial\boldsymbol{x}}\right)^T \boldsymbol{\alpha}(t,\boldsymbol{x}) - \frac{1}{2}\left(\frac{\partial\phi(t,\boldsymbol{x})}{\partial\boldsymbol{x}}\left(\frac{\partial\phi(t,\boldsymbol{x})}{\partial\boldsymbol{x}}\right)^T : \Sigma\Sigma^T\right) + \frac{1}{2}\left(\frac{\partial^2\phi(t,\boldsymbol{x})}{\partial\boldsymbol{x}^2} : \Sigma\Sigma^T\right) - \frac{1}{\tau}q\phi(t,\boldsymbol{x}) + \frac{1}{2}\left(\frac{\partial\phi(t,\boldsymbol{x})}{\partial\boldsymbol{x}}\right)^T GR^{-1}G^T\frac{\partial\phi(t,\boldsymbol{x})}{\partial\boldsymbol{x}} = 0 \tag{4.15}$$

It may be readily shown that

$$\frac{\partial\phi(t,\boldsymbol{x})}{\partial x}\left(\frac{\partial\phi(t,\boldsymbol{x})}{\partial x}\right)^T : \Sigma\Sigma^T = \frac{\partial\phi(t,\boldsymbol{x})}{\partial x}\Sigma\Sigma^T\left(\frac{\partial\phi(t,\boldsymbol{x})}{\partial x}\right)^T \tag{4.16}$$

Therefore, in order for the non-linear terms to cancel out, the following relation between τ, Σ, G, R must hold.

$$\Sigma\Sigma^T = \tau \mathrm{G}\mathrm{R}^{-1}\mathrm{G}^T \tag{4.17}$$

Finally, a linear PDE of the backward Chapman-Kolmogorov type is obtained in $\phi(t,\boldsymbol{x})$, subject to $\phi(t_f,\boldsymbol{x}) = \exp\left(-\frac{1}{\tau}g(\boldsymbol{x})\right)$.

$$-\frac{\partial\phi(t,\boldsymbol{x})}{\partial t} = \left(\frac{\partial\phi(t,\boldsymbol{x})}{\partial\boldsymbol{x}}\right)^T \boldsymbol{\alpha}(t,\boldsymbol{x}) + \frac{1}{2}\left(\frac{\partial^2\phi(t,\boldsymbol{x})}{\partial\boldsymbol{x}^2} : \Sigma\Sigma^T\right) - \frac{1}{\tau}q\phi(t,\boldsymbol{x}) = 0 \tag{4.18}$$

Thus, a log transformation of the HJB equation in conjunction with the constraint (4.17) leads to a backward Chapman-Kolmogorov type PDE. This may be solved by the Feynman-Kac formula which requires integrating the uncontrolled SDE and taking appropriate expectations. [90, 121] first proposed methods based on path integrals to solve the linear HJB equation. Later, [89] presented a more generalized approach wherein, unlike the original work, G was allowed to be a function of the state $\boldsymbol{x}$ and the scheme was flexible enough to incorporate model-free problems. Yet another way to interpret and solve the linear HJB equation given a terminal condition is through a generalized Doob's h-transform, as discussed in [122]. In this interpretation, the solution to the HJB equation may be looked upon as a probability conditioned to reach a specified terminal set. This formalism also

enables us to write an SDE, now with a suitable addition to the drift (which may be looked upon as a control term), such that the solution reaches the terminal set with probability 1. From this perspective, ϕ is nothing but the conditioned probability to reach the terminal state whilst minimizing the cost.

The second approach is to solve the HJB equation directly with the nonliner version of the Feynman-Kac formula [91]. The solution using this approach is based on a system of FBSDEs. One such scheme based on the nonlinear Feynman-Kac is described in a recently published article [93]. In the following, we first give a brief derivation of the nonlinear version of the Feynman-Kac formula, and define the FBSDEs; the reader is referred to [92] for a more comprehensive exposition. We then describe in brief the algorithm following the broad framework as in [93]. This facilitates clarity in a relative assessment of the proposed method, which uses only forward dynamics and is described in the next section.

By Ito's formula, the total differential of $v(t,\boldsymbol{X})$ may be written as

$$dv(t,\boldsymbol{X}) = \frac{\partial v(t,\boldsymbol{X})}{\partial t}dt + \left(\frac{\partial v(t,\boldsymbol{X})}{\partial \boldsymbol{x}}\right)^T \boldsymbol{\alpha}(t,\boldsymbol{X})dt + \frac{1}{2}\left(\frac{\partial^2 v(t,\boldsymbol{X})}{\partial \boldsymbol{x}^2} : \Sigma\Sigma^T\right)dt + \left(\frac{\partial v(t,\boldsymbol{X})}{\partial \boldsymbol{x}}\right)^T \Sigma d\boldsymbol{B}_t \quad (4.19)$$

Define

$$\boldsymbol{Z}(t,\boldsymbol{X}) = \Sigma^T \frac{\partial v(t,\boldsymbol{X})}{\partial \boldsymbol{x}} \quad (4.20)$$

and substitute for the first three terms from the HJB equation,

$$dv(t,\boldsymbol{X}) = -\left(q(t,\boldsymbol{X}) - \frac{1}{2}\boldsymbol{Z}(t,\boldsymbol{X})^T\Sigma^{-1}\mathrm{G}\mathrm{R}^{-1}\mathrm{G}^T\Sigma^{-T}\boldsymbol{Z}(t,\boldsymbol{X})\right)dt + \boldsymbol{Z}(t,\boldsymbol{X})^T d\boldsymbol{B}_t \quad (4.21)$$

The SDE (4.21) is subject to the terminal condition $v(t_f,\boldsymbol{X}_{t_f}) = g(\boldsymbol{X}_{t_f})$, where $\boldsymbol{X}_t = \boldsymbol{x}$. Thus, an estimate of $v(t,\boldsymbol{x})$ may be obtained by the forward integration of the uncontrolled dynamics for $\boldsymbol{X}$ from t to the final time t_f, followed by the backward

integration of v from t_f to t, such that $Y_{t_f} = g(\boldsymbol{X}_{t_f})$.

For clarity, we provide a psuedo-code for a possible scheme to implement the FBSDE as per [93], which should allow us to appreciate the benefits of the proposed scheme (see 4.3) which is based entirely on the simulation of forward SDEs.

Algorithm 2: Pseudo-code for FBSDE based SOC

Result: controlled trajectories of $\boldsymbol{X}$ and optimal control strategy $\boldsymbol{u}$
Data: $\boldsymbol{\alpha}(\boldsymbol{x},t), \Sigma, q(\boldsymbol{x},t), \mathrm{G}, \mathrm{R}, \mathrm{J}(\boldsymbol{x},t), g(\boldsymbol{x}), \Delta t$
Discretize time (assumed uniform);
$0 = t_0 < t_1 < ... < t_n = t_f \ ; \ \Delta t = t_k - t_{k-1} \forall k \in [1,n]$
Choose an ensemble size m.
for $k = 1 : n-1$ **do**
  Initialize the forward integration $\boldsymbol{X}_k^j \leftarrow \boldsymbol{X}_k \ \ \forall j \in [1,m]$
  for $p = k : n$ **do**
    for $j = 1 : m$ **do**
      $\boldsymbol{X}_{p+1}^j = \boldsymbol{X}_p^j + \boldsymbol{\alpha}(\boldsymbol{X}_p^j, t_p)\Delta t + \Sigma \Delta \boldsymbol{B}_p^j$
    end
  end
  Initialize the backward integration
  $Y_n^j \leftarrow g(\boldsymbol{X}_n^j) \ \ \forall j \in [1,m]$
  $\boldsymbol{Z}_n^j = \Sigma(t_n, \boldsymbol{X}_n^j)^T \frac{\partial g(\boldsymbol{X}_n^j)}{\partial \boldsymbol{x}} \ \ \forall j \in [1,m]$
  for $q = n-1 : k$ **do**
    $\theta_q^* =$
    $argmin_{\theta_q} \frac{1}{m} \sum_{j=1}^m \left\| \Phi(\boldsymbol{X}_q^j)\theta_q - \{Y_{q+1}^j + h(t_{q+1}, \boldsymbol{X}_{q+1}^j, \boldsymbol{Z}_{q+1}^j)\Delta t\} \right\|^2$
    where $h(t,\boldsymbol{x},\boldsymbol{z}) = \left(q(t,\boldsymbol{x}) - \frac{1}{2}\boldsymbol{z}^T \Sigma^{-1} \mathrm{G} \mathrm{R}^{-1} \mathrm{G}^T \Sigma^{-T} \boldsymbol{z}\right)$
    $Y_q^j = \Phi(\boldsymbol{X}_q^j)\theta_q^*$
    $\boldsymbol{Z}_q^j = \Sigma^T \frac{\partial \Phi(\boldsymbol{X}_q^j)}{\partial \boldsymbol{x}} \theta_q^*$
  end
  Φ represents a suitable basis function, e.g Chebyshev polynomials
  $\boldsymbol{u}_k = -\mathrm{R}^{-1}\mathrm{J}^T \boldsymbol{Z}_k$ where J is such that G$= \Sigma$J
  ($\boldsymbol{Z}_k^j \forall j \in [1,m]$ are copies of the same value, say $\boldsymbol{Z}_k$)
  $\boldsymbol{X}_{k+1} = \boldsymbol{X}_k + \boldsymbol{\alpha}(\boldsymbol{X}_k, t_k)\Delta t + \mathrm{G}\boldsymbol{u}_k \Delta t + \Sigma \Delta \boldsymbol{B}_k$
end
return $[\boldsymbol{X}_k, Y_k, \boldsymbol{Z}_k, \boldsymbol{u}_k]_{k=1}^n$

4.3 Proposed method

The basic idea behind the proposed method is to expand $g(\boldsymbol{x}_{t_f})$ using Ito's formula about a suitable time $\bar{t} < t_f$; for expositional ease we presently consider $\bar{t} = t_0$, i.e.

the initial time. We also assume that the function $g(\boldsymbol{x})$ is at least twice differentiable and define what we refer to as cost-to-reach some $\boldsymbol{x}$ at time t unlike the cost-to-go defined for the HJB equation. This is just one possible way, though not the only one, to smoothly 'smear' the Dirac spike $g(\boldsymbol{x}_t)\delta(t_f - t)$ at the terminal time over the entire time interval. While this ensures that the total cost associated with the SOC problem remains unchanged and the same control objectives are achieved, pathwise solutions thus obtained may not be the same as those obtained from the HJB equation pertaining to the original formulation. Of course, such variations in the path depend on the system dynamics being considered as the Ito expansion of $g(\boldsymbol{x}_{t_f})$ uses the generator of the stochastic differential equation. The first subsection herein has the derivation of the PDE associated with the control problem following the proposed treatment of the terminal cost. The PDE is also explicitly derived for the case of mechanical oscillators where a part of the system dynamics is not in the form of an SDE in that it contains no diffusion terms. Conforming with the same basic principle stated in the last section, for completeness, we also derive the PDE in cost-to-go in the last subsection and show that the controlled solutions obtained by both these approaches are exactly the same. This implies that the difference in the standard HJB equation and the current approach is entirely attributable to the treatment of the terminal condition via an Ito expansion about the initial time.

We specifically consider the case of LQ or $\mathscr{L}^2-$ control, where the system dynamics is linear in the control strategy and the associated cost is quadratic in it. Also, we deal with a finite time horizon for control. Adapting this work for $\mathscr{L}^1-$control and/or infinite time horizon or for any suitable stopping time poses no serious challenge, although we do not explicitly consider these scenarios here. Let the uncontrolled and controlled system dynamics be defined as in the last section, i.e. they are respectively given as

$$d\boldsymbol{X}_t = \boldsymbol{\alpha}(t,\boldsymbol{X}_t)dt + \Sigma(t,\boldsymbol{X}_t,)d\boldsymbol{B}_t \tag{4.22}$$

$$d\boldsymbol{X}_t = \boldsymbol{\alpha}(t,\boldsymbol{X}_t)dt + \mathrm{G}(t,\boldsymbol{X}_t)\boldsymbol{u}_t + \Sigma(t,\boldsymbol{X}_t)d\boldsymbol{B}_t$$

4.3.1 PDE in cost-to-reach

All functions are assumed to satisfy the smoothness and growth conditions as described in the previous section. We also define the total cost as in the previous section. However, instead of considering it as the function of the initial time, we now consider it as a function of the final time, given $\boldsymbol{X}_{t_0} := \boldsymbol{x}_0$.

$$c(t_f, \boldsymbol{x}_{t_f}, \boldsymbol{u}_{t_0 \to t_f}) = \mathbb{E}\left[g(\boldsymbol{X}_{t_f}) + \int_{t_0}^{t_f} (q(t, \boldsymbol{X}_t) + p(\boldsymbol{u}_t)) dt \right] \tag{4.23}$$

Applying Ito's formula to $g(\boldsymbol{X}_{t_f})$, we get

$$\begin{aligned} g(\boldsymbol{X}_{t_f}) &= g(\boldsymbol{X}_0) + \int_{t_0}^{t_f} \left(\frac{\partial g(\boldsymbol{X}_t)}{\partial \boldsymbol{x}} \right)^T d\boldsymbol{X}_t + \frac{1}{2} \int_{t_0}^{t_f} \left(\frac{\partial^2 g(\boldsymbol{X}_t)}{\partial \boldsymbol{x}^2} : \Sigma\Sigma^T \right) dt \\ &\qquad + \text{Ito integral} \\ &= g(\boldsymbol{X}_0) + \int_{t_0}^{t_f} \left(\mathscr{A}_t g(\boldsymbol{X}_t) + \left(\frac{\partial g(\boldsymbol{X}_t)}{\partial \boldsymbol{x}} \right)^T \mathrm{G} \boldsymbol{u}_t \right) dt + \text{Ito integral} \end{aligned} \tag{4.24}$$

where $\mathscr{A}_t$ is the generator of the original SDE equation (4.22) as given below

$$\mathscr{A}_t \xi(\boldsymbol{x}) = \sum_{i=1}^{d} \boldsymbol{\alpha}(t, \boldsymbol{x}) \frac{\partial \xi(\boldsymbol{x})}{\partial x_i} + \sum_{i=1}^{d} \sum_{j=1}^{d} \Sigma_{ij} \frac{\partial^2 \xi(\boldsymbol{x})}{\partial x_i \partial x_j} \tag{4.25}$$

Here x_i denotes the i^{th} scalar component of the vector $\boldsymbol{x}$. For a vector SDE as in (4.22), $\mathscr{A}_t \xi(\boldsymbol{x})$ may be considered as the directional derivative of the function $\xi(\boldsymbol{x})$ averaged over the paths generated by the SDE under the measure $\mathscr{P}$; see [6] for more details on the generator of an SDE. Also note that, by $\frac{\partial g(\boldsymbol{X}_t)}{\partial \boldsymbol{x}}$, we mean $\frac{\partial g(\boldsymbol{x})}{\partial \boldsymbol{x}}\big|_{\boldsymbol{x}=\boldsymbol{X}_t}$. Indeed, in lieu of the initial time t_0, we could have chosen any other deterministic or any $\mathscr{F}_t$ measurable $t^* < t_f$ to carry out the expansion above without affecting the total cost. The cost to reach $\boldsymbol{x}$ at time t, i.e. $\boldsymbol{X}_t = \boldsymbol{x}$ for some control strategy $\boldsymbol{u}$ and for a given $\boldsymbol{X}_{t_0}$, may thus be defined as

$$
\begin{aligned}
&c(t,\boldsymbol{x},\boldsymbol{u}_{t_0\to t}) \\
&\quad= \mathbb{E}\left[g(\boldsymbol{X}_{t_0}) + \int_{t_0}^{t}\left(\mathscr{A}_s g(\boldsymbol{X}_s) + \left(\frac{\partial g(\boldsymbol{X}_s)}{\partial \boldsymbol{x}}\right)^T \mathrm{G}\boldsymbol{u}_s + q(s,\boldsymbol{X}_s) + p(\boldsymbol{u}_s)\right)ds\right] \\
&\quad= \mathbb{E}\left[g(\boldsymbol{X}_{t_0}) + \int_{t_0}^{t} r(s,\boldsymbol{X}_s,\boldsymbol{u}_s)ds\right] \qquad (4.26)
\end{aligned}
$$

where $r(s,\boldsymbol{X}_s,\boldsymbol{u}_s) = \mathscr{A}_s g(\boldsymbol{X}_s) + \left(\frac{\partial g(\boldsymbol{X}_s)}{\partial \boldsymbol{x}}\right)^T \mathrm{G}\boldsymbol{u}_s + q(s,\boldsymbol{X}_s) + p(\boldsymbol{u}_s)$.

The Ito integral does not appear in the cost since its expectation is zero. Now, define the minimal cost c^* to reach some $\boldsymbol{x}$ at time t as follows.

$$
\begin{aligned}
c^*(t,\boldsymbol{x}) &= \min_{\boldsymbol{u}_{t_0\to t}} c(t,\boldsymbol{x},\boldsymbol{u}_{t_0\to t}) \qquad (4.27)\\
&= \min_{\boldsymbol{u}_{t_0\to t}} \mathbb{E}\left[g(\boldsymbol{X}_{t_0}) + \int_{t_0}^{t'} r(s,\boldsymbol{X}_s,\boldsymbol{u}_s)ds + \int_{t'}^{t} r(s,\boldsymbol{X}_s,\boldsymbol{u}_s)ds\right] \\
&= \min_{\boldsymbol{u}_{t'\to t}} \mathbb{E}\left[\min_{\boldsymbol{u}_{t_0\to t'}} \mathbb{E}\left[g(\boldsymbol{X}_{t_0}) + \int_{t_0}^{t'} r(s,\boldsymbol{X}_s,\boldsymbol{u}_s)dt\right] + \int_{t'}^{t} r(s,\boldsymbol{X}_s,\boldsymbol{u}_s)dt\right] \\
&= \min_{\boldsymbol{u}_{t'\to t}} \mathbb{E}\left[c^*(t',\boldsymbol{X}_{t'}) + \int_{t'}^{t} r(s,\boldsymbol{X}_s,\boldsymbol{u}_s)dt\right]
\end{aligned}
$$

Let $t' = t - dt$. Upon Ito-Taylor expansion of $c^*(t',\boldsymbol{x}(t'))$ around $(t,\boldsymbol{x})$, we have

$$
\begin{aligned}
&\mathbb{E}[c^*(t-dt,\boldsymbol{X}_{t-dt})|\boldsymbol{X}_t = \boldsymbol{x}] \\
&= \mathbb{E}\left[c^*(t,\boldsymbol{x}) - \frac{\partial c^*(t,\boldsymbol{x})}{\partial t}dt - \left(\frac{\partial c^*(t,\boldsymbol{x})}{\partial \boldsymbol{x}}\right)^T d\boldsymbol{X}_t - \frac{1}{2}\left(\frac{\partial^2 c^*(t,\boldsymbol{x})}{\partial \boldsymbol{x}^2} : d\boldsymbol{X}_t d\boldsymbol{X}_t^T\right)\right] \\
&= c^*(t,\boldsymbol{x}) - \frac{\partial c^*(t,\boldsymbol{x})}{\partial t}dt - \left(\frac{\partial c^*(t,\boldsymbol{x})}{\partial \boldsymbol{x}}\right)^T (\boldsymbol{\alpha}(t,\boldsymbol{x}) + \mathrm{G}\boldsymbol{u})dt \\
&\qquad - \frac{1}{2}\left(\frac{\partial^2 c^*(t,\boldsymbol{x})}{\partial \boldsymbol{x}^2} : \Sigma\Sigma^T\right)dt \qquad (4.28)
\end{aligned}
$$

Substituting this in equation (4.27), we have

$$
\begin{aligned}
\frac{\partial c^*(t,\boldsymbol{x})}{\partial t} = \min_{\boldsymbol{u}_t}\Bigg[&- \left(\frac{\partial c^*(t,\boldsymbol{x})}{\partial \boldsymbol{x}}\right)^T (\boldsymbol{\alpha}(t,\boldsymbol{x}) + \mathrm{G}\boldsymbol{u}_t) - \frac{1}{2}\left(\frac{\partial^2 c^*(t,\boldsymbol{x})}{\partial \boldsymbol{x}^2} : \Sigma\Sigma^T\right) \\
&+ r(t,\boldsymbol{x},\boldsymbol{u}_t)\Bigg] \qquad (4.29)
\end{aligned}
$$

$$\frac{\partial c^*(t,\boldsymbol{x})}{\partial t} = \min_{\boldsymbol{u}_t}\Bigg[-\left(\frac{\partial c^*(t,\boldsymbol{x})}{\partial \boldsymbol{x}}\right)^T(\boldsymbol{\alpha}(t,\boldsymbol{x}) + \mathrm{G}\boldsymbol{u}_t) - \frac{1}{2}\left(\frac{\partial^2 c^*(t,\boldsymbol{x})}{\partial \boldsymbol{x}^2} : \Sigma\Sigma^T\right) + \mathscr{A}_t g(\boldsymbol{x}) + \left(\frac{\partial g(\boldsymbol{x})}{\partial \boldsymbol{x}}\right)^T \mathrm{G}\boldsymbol{u}_t + q(t,\boldsymbol{x}) + p(\boldsymbol{u}_t)\Bigg] \quad (4.30)$$

Since the quantity to be minimized is quadratic in $\boldsymbol{u}_t$, an analytical expression for the optimal $\boldsymbol{u}_t = \boldsymbol{u}_t^*$ may be obtained as follows.

$$\boldsymbol{u}^*(t,\boldsymbol{x}) = \mathrm{R}^{-1}\mathrm{G}^T\left(\frac{\partial c^*(t,\boldsymbol{x})}{\partial \boldsymbol{x}} - \frac{\partial g(\boldsymbol{x})}{\partial \boldsymbol{x}}\right) \quad (4.31)$$

Plugging the above into equation (4.30), and simplifying, we obtain the PDE in optimal cost-to-reach as follows.

$$\begin{aligned}\frac{\partial c^*(t,\boldsymbol{x})}{\partial t} &+ \left(\frac{\partial c^*(t,\boldsymbol{x})}{\partial \boldsymbol{x}}\right)^T\boldsymbol{\alpha}(t,\boldsymbol{x}) + \frac{1}{2}\left(\frac{\partial^2 c^*(t,\boldsymbol{x})}{\partial \boldsymbol{x}^2} : \Sigma\Sigma^T\right) \\ &- \left(\frac{\partial c^*(t,\boldsymbol{x})}{\partial \boldsymbol{x}}\right)^T \mathrm{G}\mathrm{R}^{-1}\mathrm{G}^T\frac{\partial g(\boldsymbol{x})}{\partial \boldsymbol{x}} + \frac{1}{2}\left(\frac{\partial c^*(t,\boldsymbol{x})}{\partial \boldsymbol{x}}\right)^T \mathrm{G}\mathrm{R}^{-1}\mathrm{G}^T\frac{\partial c^*(t,\boldsymbol{x})}{\partial \boldsymbol{x}} \\ &+ \frac{1}{2}\left(\frac{\partial g(\boldsymbol{x})}{\partial \boldsymbol{x}}\right)^T \mathrm{G}\mathrm{R}^{-1}\mathrm{G}^T\frac{\partial g(\boldsymbol{x})}{\partial \boldsymbol{x}} - \left(\frac{\partial g(\boldsymbol{x})}{\partial \boldsymbol{x}}\right)^T\boldsymbol{\alpha}(t,\boldsymbol{x}) - \frac{1}{2}\left(\frac{\partial^2 g(\boldsymbol{x})}{\partial \boldsymbol{x}^2} : \Sigma\Sigma^T\right) \\ &- q(t,\boldsymbol{x}) = 0 \ ; \ \text{subject to } c^*(t_0,\boldsymbol{x}_0) = g(\boldsymbol{x}_0) \end{aligned} \quad (4.32)$$

We now proceed to determine the set of SDEs to solve for c^*. Let us start by defining the total derivative of c^* based on Ito's formula.

$$\begin{aligned} dc^*(t,\boldsymbol{X}_t) = \frac{\partial c^*(t,\boldsymbol{X}_t)}{\partial t}dt &+ \left(\frac{\partial c^*(t,\boldsymbol{X}_t)}{\partial \boldsymbol{x}}\right)^T\boldsymbol{\alpha}(t,\boldsymbol{X}_t)dt \\ &+ \frac{1}{2}\left(\frac{\partial^2 c^*(t,\boldsymbol{X}_t)}{\partial \boldsymbol{x}^2} : \Sigma\Sigma^T\right)dt + \left(\frac{\partial c^*(t,\boldsymbol{X}_t)}{\partial \boldsymbol{x}}\right)^T\Sigma d\boldsymbol{B}_t \end{aligned} \quad (4.33)$$

As is conventional in the FBSDE literature, we define

$$\boldsymbol{Z}_t = \Sigma^T\frac{\partial c^*(t,\boldsymbol{X}_t)}{\partial \boldsymbol{x}} \quad (4.34)$$

Also, to ensure the absolute continuity of measures corresponding to the controlled

and uncontrolled dynamics, the following relation must hold for some J, presently assumed to be Lipschitz continuous and with a sublinear growth bound.

$$\mathrm{G}(t,\boldsymbol{X}_t) = \Sigma \mathrm{J}(t,\boldsymbol{X}_t) \tag{4.35}$$

Using the PDE (4.32), we substitute for the first three terms in the total derivative in equation (4.33). Finally, making use of equations (4.34) and (4.35), we have an SDE in c^* as follows:

$$dc^*(t,\boldsymbol{X_t}) = -\tilde{h}(\boldsymbol{X}_t, c^*_t, \boldsymbol{Z}_t)dt + \boldsymbol{Z}_t^T d\boldsymbol{B}_t \;\; ; \;\; c^*(t_0,\boldsymbol{X}_{t_0}) = g(\boldsymbol{X}_{t_0}) \tag{4.36}$$

where

$$\begin{aligned}\tilde{h}(\boldsymbol{X_t}, c^*, \boldsymbol{Z}_t) = &-\boldsymbol{Z}_t^T \mathrm{J}\mathrm{R}^{-1}\mathrm{G}^T \frac{\partial g(\boldsymbol{X}_t)}{\partial \boldsymbol{x}} + \frac{1}{2}\boldsymbol{Z}_t^T \mathrm{J}\mathrm{R}^{-1}\mathrm{J}^T \boldsymbol{Z}_t \\ &+ \frac{1}{2}\left(\frac{\partial g(\boldsymbol{X}_t)}{\partial \boldsymbol{x}}\right)^T \mathrm{G}\mathrm{R}^{-1}\mathrm{G}^T \frac{\partial g(\boldsymbol{X}_t)}{\partial \boldsymbol{x}} - \left(\frac{\partial g(\boldsymbol{X}_t)}{\partial \boldsymbol{x}}\right)^T \alpha(t,\boldsymbol{X}_t) \\ &- \frac{1}{2}\left(\frac{\partial^2 g(\boldsymbol{X}_t)}{\partial \boldsymbol{x}^2} : \Sigma\Sigma^T\right) - q(t,\boldsymbol{X}_t)\end{aligned} \tag{4.37}$$

Thus, we have a set of SDEs for the uncontrolled dynamics for $\boldsymbol{X}$ and the cost-to-reach function c^* which is subject to the initial condition $g(\boldsymbol{X}_{t_0})$. We can simulate these two SDEs simultaneously to obtain the optimal control strategy $\boldsymbol{u}_t^*$ at any time as per equation (4.31). A pseudo-code is provided below for clarity. In what follows, we replace c^* by Y, to be consistent with the typical notation used in the literature.

4.3.2 Control scheme for mechanical oscillators

For ease of exposition, we present the scheme for a single-degree-of-freedom (SDOF) system and note that an extension to a multi-degree-of-freedom (MDOF) system is straightforward. Thus, consider the following general form of an SDOF mechanical oscillator.

$$\ddot{\zeta} + \alpha(\zeta,\dot{\zeta}) = \beta(t) + \sigma(\zeta,t)W \tag{4.38}$$

Algorithm 3: Pseudo-code for proposed method

Result: controlled trajectories of the state $\boldsymbol{X}$ and optimal control $\boldsymbol{u}$

Data: $\boldsymbol{\alpha}(t,\boldsymbol{x}), \Sigma, q(t,\boldsymbol{x}), \mathrm{G}, \mathrm{R}, \mathrm{J}(\boldsymbol{x},t), g(\boldsymbol{x}), \Delta t, s, h$

Discretize time (assumed uniform);

$0 = t_0 < t_1 < \ldots < t_n = t_f \;\; ; \;\; \Delta t = t_k - t_{k-1} \forall k \in [1,n]$

Choose an ensemble size m.

Initialize:

$\boldsymbol{X}_0$

$\boldsymbol{Y}_0 \leftarrow g(\boldsymbol{X}_0)$

$\boldsymbol{Z}_0 \leftarrow \Sigma^T \frac{\partial g(\boldsymbol{x})}{\partial \boldsymbol{x}}\big|_{\boldsymbol{x}=\boldsymbol{X_0}}$

$\boldsymbol{u}_0 \leftarrow \mathrm{R}^{-1}\mathrm{G}^T \left(\Sigma^{-T}\boldsymbol{Z}_0 - \frac{\partial g(\boldsymbol{x})}{\partial \boldsymbol{x}}\big|_{\boldsymbol{x}=\boldsymbol{X_0}}\right)$

for $k = 1 : n$ **do**

  for $j = 1 : m$ **do**

    $\boldsymbol{X}_k^j = \boldsymbol{X}_{k-1} + \boldsymbol{\alpha}(t_{k-1}, \boldsymbol{X}_{k-1}) + \mathrm{G}\boldsymbol{u}_{k-1}\Delta t + \Sigma \Delta \boldsymbol{B}_{k-1}^j$

    Integrate Y implicitly in $\boldsymbol{X}$ and expicitly in $Y, \boldsymbol{Z}$

    $Y_k^j = Y_{k-1} - \tilde{h}(\boldsymbol{X}_k^j, Y_{k-1}, \boldsymbol{Z}_{k-1})\Delta t + \boldsymbol{Z}_{k-1}^T \Delta \boldsymbol{B}_{k-1}^j$

    where

    $\tilde{h}(\boldsymbol{X}_k^j, Y_{k-1}, \boldsymbol{Z}_{k-1}) = -q(t_{k-1}, \boldsymbol{X}_k^j) - \boldsymbol{Z}_{k-1}^T \mathrm{J}\mathrm{R}^{-1}\mathrm{G}^T \frac{\partial g(\boldsymbol{x})}{\partial \boldsymbol{x}}\big|_{\boldsymbol{x}=\boldsymbol{X}_k^j}$

    $+\frac{1}{2}\boldsymbol{Z}_{k-1}^T \mathrm{J}\mathrm{R}^{-1}\mathrm{J}^T \boldsymbol{Z}_{k-1} + \frac{1}{2}\left(\frac{\partial g(\boldsymbol{x})}{\partial \boldsymbol{x}}\big|_{\boldsymbol{x}=\boldsymbol{X}_k^j}\right)^T \mathrm{G}\mathrm{R}^{-1}\mathrm{G}^T \left(\frac{\partial g(\boldsymbol{x})}{\partial \boldsymbol{x}}\big|_{\boldsymbol{x}=\boldsymbol{X}_k^j}\right)$

    $-\left(\frac{\partial g(\boldsymbol{x})}{\partial \boldsymbol{x}}\big|_{\boldsymbol{x}=\boldsymbol{X}_k^j}\right)^T \boldsymbol{\alpha}(t, \boldsymbol{X}_k^j) - \frac{1}{2}\left(\frac{\partial^2 g(\boldsymbol{x})}{\partial \boldsymbol{x}^2}\big|_{\boldsymbol{x}=\boldsymbol{X}_k^j} : \Sigma\Sigma^T\right)$

  end

  $\boldsymbol{X}_k \leftarrow \frac{1}{m}\sum_{j=1}^m \boldsymbol{X}_k^j$

  Fit the ensemble of Y_k in terms of Gaussian basis functions $\eta_{(i)}$ of $\boldsymbol{X}_k$

  to obtain $\boldsymbol{\theta}_k^*$ such that $\eta_{(i)}(\boldsymbol{x}) = \exp\left(-\frac{(\boldsymbol{x}-\boldsymbol{x}_i)^T(\boldsymbol{x}-\boldsymbol{x}_i)}{2s^2}\right)$ where

  $\boldsymbol{x}_i \in \{\boldsymbol{X}_k \cup$ nodes on the discretization cells (each of size h) around

  $\boldsymbol{X}_k\}$; s, h are parameters of the Gaussian basis

  $\boldsymbol{\theta}_k^* = argmin_{\boldsymbol{\theta}} \frac{1}{m}\sum_{j=1}^m \left\| Y_k^j - \boldsymbol{\eta}(\boldsymbol{X}_k)^T \boldsymbol{\theta} \right\|^2$

  ($\boldsymbol{\eta}$ is a vector valued function with components $\eta_{(i)}$)

  $Y_k \leftarrow \boldsymbol{\eta}(\boldsymbol{X}_k)^T \boldsymbol{\theta}_k^*$

  $\boldsymbol{Z}_k = \Sigma^T \frac{\partial \boldsymbol{\eta}^T(\boldsymbol{x})}{\partial \boldsymbol{x}}\big|_{\boldsymbol{x}=\boldsymbol{X_k}} \boldsymbol{\theta}_k^*.$

  $\boldsymbol{u}_k = \mathrm{R}^{-1}\mathrm{G}^T \left(\Sigma^{-T}\boldsymbol{Z}_k - \frac{\partial g(\boldsymbol{x})}{\partial \boldsymbol{x}}\big|_{\boldsymbol{x}=\boldsymbol{X_k}}\right)$

end

return $[\boldsymbol{X}_k, Y_k, \boldsymbol{Z}_k, \boldsymbol{u}_k]_{k=1}^n$

We assume $\alpha(\zeta,\dot{\zeta})$, $\sigma(\zeta,t)$ and $\beta(t)$ to be C^1, $C^{2,1}$ and C^1 in their respective arguments. $W=\dot{B}$ in the equation above formally refers to a white noise process and in order to make the equation more tractable, it may be recast as a system of two first order equations as follows.

$$\begin{Bmatrix} dX_1 \\ dX_2 \end{Bmatrix} = \begin{Bmatrix} X_2 \\ -\alpha(X_1,X_2)+\beta(t) \end{Bmatrix} dt + \begin{bmatrix} 0 & 0 \\ 0 & \sigma(X_1,t) \end{bmatrix} \begin{Bmatrix} 0 \\ dB_t \end{Bmatrix} \tag{4.39}$$

where $X_1 := \zeta, X_2 := \dot{\zeta}$. This system is inherently different from the one studied earlier wherein the system dynamics was represented by a system of SDEs, each with a nonzero diffusion term. Presently however, for a d-DOF mechanical oscillator, the state space representation is in a $2d$-dimensional space wherein the first d equations are deterministic (i.e. without diffusion terms) and the remaining d are stochastic. For the sake of brevity, we continue with the derivation of the PDE for an SDOF oscillator, i.e. when $d=1$, although it may be easily adapted to MDOF systems. The equation of motion for a linearly controlled SDOF system may be written as

$$\begin{Bmatrix} dX_1 \\ dX_2 \end{Bmatrix} = \begin{Bmatrix} X_2 \\ -\alpha(X_1,X_2)+\beta(t)+\mathrm{G}u_t \end{Bmatrix} dt + \begin{bmatrix} 0 & 0 \\ 0 & \sigma(X_1,t) \end{bmatrix} \begin{Bmatrix} 0 \\ dB_t \end{Bmatrix} \tag{4.40}$$

Note that, G here is a scalar. For the system represented by (4.39), the infinitesimal generator of the original (uncontrolled) system acting on an appropriately smooth function $f(x_1,x_2) \equiv f(\boldsymbol{x})$ may be written as

$$\mathscr{A}_t f(\boldsymbol{x}) = x_2 \frac{\partial f(\boldsymbol{x})}{\partial x_1} + (-\alpha(\boldsymbol{x})+\beta(t)) \frac{\partial f(\boldsymbol{x})}{\partial x_2} + \frac{\sigma^2}{2} \frac{\partial^2 f(\boldsymbol{x})}{\partial x_2^2} \tag{4.41}$$

where $\boldsymbol{x} := [x_1,x_2]^T$. As in the previous section, the cost-to-reach some point $\boldsymbol{x} \in \mathbb{R}^2$ at time t for some control strategy u is

$$c(t,\boldsymbol{x},u_{t_0\to t}) = \mathbb{E}\left[g(\boldsymbol{x}_0) + \int_{t_0}^{t} [\mathscr{A}_s g(\boldsymbol{x}_s) + \frac{\partial g}{\partial x_2}\mathrm{G}u_t + q(\boldsymbol{x}_s) + p(u_s)]ds \right] \tag{4.42}$$

where the definition of $\mathscr{A}_t$ is given by equation (4.41) and $p(u_t) = \frac{\mathrm{R}u_t^2}{2}$ for some scalar R. From here onwards, proceeding exactly as in the previous subsection, we arrive at the following PDE and optimal control strategy.

$$\frac{\partial c^*(t,\boldsymbol{x})}{\partial t} + x_2\frac{\partial c^*(t,\boldsymbol{x})}{\partial x_1} + (-\alpha(\boldsymbol{x}) + \beta(t))\frac{\partial c^*(t,\boldsymbol{x})}{\partial x_2} + \frac{\sigma^2}{2}\frac{\partial^2 c^*(t,\boldsymbol{x})}{\partial x_2^2} - \frac{\mathrm{G}^2}{2\mathrm{R}}\left(\frac{\partial c^*(t,\boldsymbol{x})}{\partial x_2} - \frac{\partial g(\boldsymbol{x})}{\partial x_2}\right)^2 - q(\boldsymbol{x}) - x_2\frac{\partial g(\boldsymbol{x})}{\partial x_1} - (-\alpha(\boldsymbol{x}) + \beta(t))\frac{\partial g(\boldsymbol{x})}{\partial x_2} - \frac{\sigma^2}{2}\frac{\partial^2 g(\boldsymbol{x})}{\partial x_2^2} = 0 \ ; \ \text{subject to } c^*(t_0,\boldsymbol{x}) = g(\boldsymbol{x}) \quad (4.43)$$

$$u^*(t,\boldsymbol{x}) = \frac{\mathrm{G}}{\mathrm{R}}\left(\frac{\partial c^*(t,\boldsymbol{x})}{\partial x_2} - \frac{\partial g(\boldsymbol{x})}{\partial x_2}\right) \quad (4.44)$$

Thus the SDE for $c^*(t,\boldsymbol{x})$ is

$$dc^*(t,\boldsymbol{X}_t) := dY_t = \left(-\mathscr{A}_t g(\boldsymbol{X}_t) - q(\boldsymbol{X}_t) + \frac{\mathrm{G}^2}{2\sigma^2\mathrm{R}}\left(Z_t - \sigma\frac{\partial g(\boldsymbol{X}_t)}{\partial x_2}\right)^2\right)dt + Z_t dB_t \quad (4.45)$$

$$\text{where } Z_t = \sigma\frac{\partial c^*(t,\boldsymbol{x})}{\partial x_2}\bigg|_{\boldsymbol{x}=\boldsymbol{X}_t} \quad (4.46)$$

With the state dynamics given by equation (4.40), and the drift $\tilde{h}$ as defined below for the SDE in Y, Algorithm 3 may be used for the control of oscillators whose dynamics are of the form (4.38).

$$\tilde{h}(\boldsymbol{x},z) = x_2\frac{\partial g(\boldsymbol{x})}{\partial x_1} + \left(-\alpha(\boldsymbol{x}) + \beta(t)\right)\frac{\partial g(\boldsymbol{x})}{\partial x_2} + \frac{\sigma^2}{2}\frac{\partial^2 g(\boldsymbol{x})}{\partial x_2^2} + q(\boldsymbol{x}) - \frac{\mathrm{G}^2}{2\sigma^2\mathrm{R}}\left(z - \sigma\frac{\partial g(\boldsymbol{x})}{\partial x_2}\right)^2 \quad (4.47)$$

4.3.3 Derivation in cost-to-go

Following the definition of the total cost from the previous subsection, the cost-to-go (denoted $\hat{c}$) to some final state starting at $\boldsymbol{x}$ and at time t may be written as

$$\hat{c}(t,\boldsymbol{x}) = \min_{\boldsymbol{u}_{t\to t_f}} \mathbb{E}\left[\int_t^{t_f} r(s,\boldsymbol{X}_s,\boldsymbol{u}_s)ds \middle| \boldsymbol{X}_t = \boldsymbol{x}\right] \tag{4.48}$$

$$\hat{c}(t',\boldsymbol{x}') = \min_{\boldsymbol{u}_{t'\to t_f}} \mathbb{E}\left[\int_{t'}^{t_f} r(s,\boldsymbol{X}_s,\boldsymbol{u}_s)ds \middle| \boldsymbol{X}_{t'} = \boldsymbol{x}'\right] \tag{4.49}$$

$$r(s,\boldsymbol{x}_s,\boldsymbol{u}_s) = \mathscr{A}_s g(\boldsymbol{x}) + \left(\frac{\partial g(\boldsymbol{x})}{\partial \boldsymbol{x}}\right)^T \mathrm{G}\boldsymbol{u_s} + q(s,\boldsymbol{x}_s) + p(\boldsymbol{u}_s)$$

$\mathscr{A}_s$: generator of the uncontrolled SDE for $\boldsymbol{X}$

$$\hat{c}(t,\boldsymbol{x}) = \min_{\boldsymbol{u}_{t\to t'}} \mathbb{E}\left[\left(\int_t^{t'} r(s,\boldsymbol{X}_s,\boldsymbol{u}_s)ds + \min_{\boldsymbol{u}_{t'\to t_f}} \mathbb{E}\left(\int_{t'}^{t_f} r(s,\boldsymbol{X}_s,\boldsymbol{u}_s)ds \middle| \boldsymbol{X}_{t'} = \boldsymbol{x}'\right)\right) \middle| \boldsymbol{X}_t = \boldsymbol{x}\right] \tag{4.50}$$

$$\hat{c}(t,\boldsymbol{x}) = \min_{\boldsymbol{u}_{t\to t'}} \mathbb{E}\left[\int_t^{t'} r(s,\boldsymbol{X}_s,\boldsymbol{u}_s)ds + \hat{c}(t',\boldsymbol{x}') \middle| \boldsymbol{X}_t = \boldsymbol{x}\right] \tag{4.51}$$

For $t' = t + dt$

$$\hat{c}(t,\boldsymbol{x}) = \min_{\boldsymbol{u}_{t\to t+dt}} \mathbb{E}\left[r(t,\boldsymbol{X}_t,\boldsymbol{u}_t)dt + \hat{c}(t+dt,\boldsymbol{X}_{t+dt}) \middle| \boldsymbol{X}_t = \boldsymbol{x}\right] \tag{4.52}$$

$$\mathbb{E}[\hat{c}(t+dt,\boldsymbol{X}_{t+dt})|\boldsymbol{X}_t = \boldsymbol{x}] = \hat{c}(t,\boldsymbol{x}) + \frac{\partial \hat{c}(t,\boldsymbol{x})}{\partial t}dt + \left(\frac{\partial \hat{c}(t,\boldsymbol{x})}{\partial \boldsymbol{x}}\right)^T \boldsymbol{\alpha}(t,\boldsymbol{x})dt + \left(\frac{\partial \hat{c}(t,\boldsymbol{x})}{\partial \boldsymbol{x}}\right)^T \mathrm{G}\boldsymbol{u}dt + \frac{1}{2}\left(\frac{\partial^2 \hat{c}(t,\boldsymbol{x})}{\partial \boldsymbol{x}^2} : \Sigma\Sigma^T\right)dt \tag{4.53}$$

$$\therefore -\frac{\partial \hat{c}(t,\boldsymbol{x})}{\partial t} = r(t,\boldsymbol{x},\boldsymbol{u}) + \left(\frac{\partial \hat{c}(t,\boldsymbol{x})}{\partial \boldsymbol{x}}\right)^T \boldsymbol{\alpha}(t,\boldsymbol{x}) + \left(\frac{\partial \hat{c}(t,\boldsymbol{x})}{\partial \boldsymbol{x}}\right)^T \mathrm{G}\boldsymbol{u} + \frac{1}{2}\left(\frac{\partial^2 \hat{c}(t,\boldsymbol{x})}{\partial \boldsymbol{x}^2} : \Sigma\Sigma^T\right) \tag{4.54}$$

$$= \mathscr{A}_t \hat{c} + \left(\frac{\partial \hat{c}(t,\boldsymbol{x})}{\partial \boldsymbol{x}}\right)^T \mathrm{G}\boldsymbol{u} + q(t,\boldsymbol{x}) + p(\boldsymbol{u}) + \mathscr{A}_t g(\boldsymbol{x}) + \left(\frac{\partial g(\boldsymbol{x})}{\partial \boldsymbol{x}}\right)^T \mathrm{G}\boldsymbol{u} \tag{4.55}$$

$$\boldsymbol{u}^*(t,\boldsymbol{x}) = -\mathrm{R}^{-1}\mathrm{G}^T\left(\frac{\partial \hat{c}(t,\boldsymbol{x})}{\partial \boldsymbol{x}} + \frac{\partial g(\boldsymbol{x})}{\partial \boldsymbol{x}}\right) \tag{4.56}$$

Substituting $\boldsymbol{u}^*$ in the PDE in $\hat{c}$ and simplifying, we obtain, as in the case of the HJB equation, a PDE in cost-to-go subject to a terminal condition. In this case, however $\hat{c}(t_f, \boldsymbol{x}_{t_f}) = 0$ on account of the Ito-expansion of the terminal cost, unlike the HJB equation where the terminal cost is $g(\boldsymbol{x}_{t_f})$.

$$\frac{\partial \hat{c}(t,\boldsymbol{x})}{\partial t} + \mathscr{A}_t \hat{c}(t,\boldsymbol{x}) + q(t,\boldsymbol{x}_t) + \mathscr{A}_t g(\boldsymbol{x}) - \frac{1}{2}\left(\frac{\partial \hat{c}(t,\boldsymbol{x})}{\partial \boldsymbol{x}} + \frac{\partial g(\boldsymbol{x})}{\partial \boldsymbol{x}}\right)^T \mathrm{G}\mathrm{R}^{-1}\mathrm{G}^T \left(\frac{\partial \hat{c}(t,\boldsymbol{x})}{\partial \boldsymbol{x}} + \frac{\partial g(\boldsymbol{x})}{\partial \boldsymbol{x}}\right) = 0 \quad (4.57)$$

$$d\hat{c}(t,\boldsymbol{X}) = \frac{\partial \hat{c}(t,\boldsymbol{X})}{\partial t} + \mathscr{A}_t \hat{c}(t,\boldsymbol{X}) + \left(\frac{\partial \hat{c}(t,\boldsymbol{X})}{\partial \boldsymbol{X}}\right)^T \Sigma d\boldsymbol{B}_t \quad (4.58)$$

$$\begin{aligned} d\hat{c}(t,\boldsymbol{X}) = & -q(t,\boldsymbol{X}_t) - \mathscr{A}_t g(\boldsymbol{X}) \\ & - \frac{1}{2}\left(\frac{\partial \hat{c}(t,\boldsymbol{X})}{\partial \boldsymbol{x}} + \frac{\partial g(\boldsymbol{X})}{\partial \boldsymbol{x}}\right)^T \mathrm{G}\mathrm{R}^{-1}\mathrm{G}^T \left(\frac{\partial \hat{c}(t,\boldsymbol{X})}{\partial \boldsymbol{x}} + \frac{\partial g(\boldsymbol{X})}{\partial \boldsymbol{x}}\right) \\ & + \left(\frac{\partial \hat{c}(t,\boldsymbol{X})}{\partial \boldsymbol{x}}\right)^T \Sigma d\boldsymbol{B}_t \end{aligned} \quad (4.59)$$

Now, since the optimal cost-to-go and cost-to-reach are complements of each other, their sum may be considered as constant (i.e. a random variable with time-independent expectations). Let the total cost be c_{t_f}, so that we have $\hat{c} = c_{t_f} - c^*$. Performing the change of variable, the previous SDE in $\hat{c}$ may be re-written as an SDE in c^* as follows.

$$\begin{aligned} -dc^*(t,\boldsymbol{X}) = & -\frac{1}{2}\left(-\frac{\partial c^*(t,\boldsymbol{X})}{\partial \boldsymbol{x}} + \frac{\partial g(\boldsymbol{X})}{\partial \boldsymbol{x}}\right)^T \mathrm{G}\mathrm{R}^{-1}\mathrm{G}^T \left(-\frac{\partial c^*(t,\boldsymbol{X})}{\partial \boldsymbol{x}} + \frac{\partial g(\boldsymbol{X})}{\partial \boldsymbol{x}}\right) \\ & - q(t,\boldsymbol{X}) - \mathscr{A}_t g(\boldsymbol{X}) - \left(\frac{\partial c^*(t,\boldsymbol{X})}{\partial \boldsymbol{x}}\right)^T \Sigma d\boldsymbol{B}_t \end{aligned} \quad (4.60)$$

$$\begin{aligned} dc^*(t,\boldsymbol{X}) = & -\frac{1}{2}\left(-\frac{\partial c^*(t,\boldsymbol{X})}{\partial \boldsymbol{x}} + \frac{\partial g(\boldsymbol{X})}{\partial \boldsymbol{x}}\right)^T \mathrm{G}\mathrm{R}^{-1}\mathrm{G}^T \left(-\frac{\partial c^*(t,\boldsymbol{X})}{\partial \boldsymbol{x}} + \frac{\partial g(\boldsymbol{X})}{\partial \boldsymbol{x}}\right) \\ & + q(t,\boldsymbol{X}) + \mathscr{A}_t g(\boldsymbol{X}) + \left(\frac{\partial c^*(t,\boldsymbol{X})}{\partial \boldsymbol{x}}\right)^T \Sigma d\boldsymbol{B}_t \end{aligned} \quad (4.61)$$

This SDE is exactly the same as that obtained in Section 4.3.1. It was possible to obtain the SDE in c^* from that of $\hat{c}$ without knowing the total cost c_T since the drift and diffusion terms of $\hat{c}$ were only functions of its derivatives and not of the function itself.

4.4 Numerical illustrations

In this section, we implement the proposed method for the control of a class of stochastic dynamical systems with emphasis on mechanical oscillators. There is no restriction on the model dynamics except that the control term is linear and its associated cost quadratic. However, before moving on to the examples, a word about the relative computational costs for the FBSDE based and current methods. Assuming a uniform step size Δt, let n be the total number of time steps such that $n = \frac{t_f - t_0}{\Delta t}$. Then, in the FBSDE method, the uncontrolled forward dynamics and the backward dynamics must be integrated $\frac{n^2+n}{2}$ times each. Contrast it with the present method requiring just n integrations for the SDE in Y and the same for the controlled SDE in X. The FBSDE method also requires n integration steps of the controlled dynamics which need not be done explicitly in the present method since the SDEs move only forward in time. Perhaps the most significant computational overhead is in the function approximations that must be performed every integration step for the SDE in Y. For the FBSDE method, this costly evaluation must be undertaken $\frac{n^2+n}{2}$ times in contrast with just n such evaluations in the present scheme. Longer integration times for computing the optimal control at every step also imply substantially higher accumulation of integration errors and hence higher sampling variance too in the FBSDE method (see Figure 4.1). Controlling this substantial source of error in the FBSDE would necessitate the use of higher order numerical integration schemes for the SDEs, which would add to the computational burden. Unlike the FBSDE, the present approach also affords the freedom to choose (or even optimize) the initial time $\bar{t}$ to Ito-expand the terminal cost function $g(\boldsymbol{x})$; see subsequent remarks during our numerical work on the control of chaos. Finally, with our proposed scheme, simulations of the SDEs in X and Y could be done in

parallel and this is not possible with the FBSDE scheme.

We start with the control of a simple 1-dimensional problem which enables a ready comparison with a closed-form solution. The second example is on a 2-dimensional Ornstein-Uhlenbeck process which is of interest in studying phase modulations in the oscillatory dynamics encountered in myriad physical, mechanical and biological systems [123]. Next, we study the control of the 3-dimensional Lorenz 1963 model, where the uncontrolled dynamics is in the chaotic regime. Finally, we consider the control of a couple of nonlinear mechanical oscillators, viz. hardening Duffing and Duffing-Holmes, involving chaotic as well as nonchaotic behaviour.

A 1D system

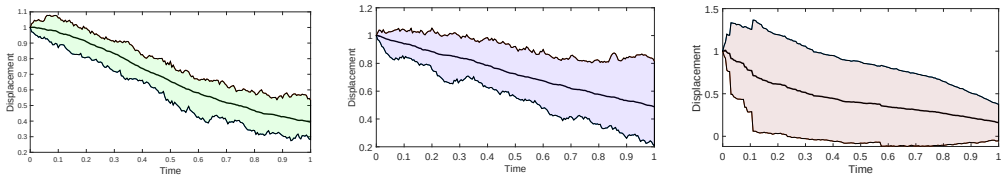
In our first example, a simple 1D problem is considered. The equation of motion is given by the following SDE.

$$dX_t = 0.2X_t dt + 0.5dB_t \tag{4.62}$$

The terminal cost is given by the function $g(x) = wx^2$. No running cost on X is assumed while that on u is quadratic and of the form $\frac{\mathrm{R}}{2}u^2$. Hence, the total cost is

$$c = \{wx_{t_f}^2 + \int_0^{t_f} \frac{\mathrm{R}}{2}u^2\}, w = 10, \mathrm{R} = 2 \tag{4.63}$$

For this example, the optimal control u_t may be found analytically by solving a Riccati type ODE; see [5]. Figure 4.1 shows the mean controlled path over 50 independent simulations of the proposed method as well as that pertaining to the closed-form solution, each with an ensemble size (m) of 10. Also plotted is the solution based on an FBSDE scheme for the same ensemble size. As is clear from the figures, the sampling variance with the proposed method is the least. The total mean cost determined by numerically integrating the paths (see equation 4.63) along with its variance across independent simulations is also reported in Table 4.1. Of particular interest is the comparison between the FBSDE and proposed methods since closed form solution is rarely available.



(a) Proposed method **(b)** Theoretical solution **(c)** FBSDE method

Figure 4.1: Comparison of results from various methods for the 1D problem. In each figure, black bold line shows the mean and colored patch represents the variation in performance over 50 independent simulations; i.e. at any time, the boundary of the colored patch represents the minimum and maximum displacement at that time. Time is measured in seconds and displacement in meters.

Method	**Mean of total cost**	**Variance of total cost**	**Computational time (sec)**
Proposed method	4.4579	1.5578	30
Closed form solution	4.3422	2.8674	0.028
FBSDE[1] method	327.5918	4.65×10^5	605

Table 4.1: A comparison of costs and computational time

A 2-D Ornstein-Uhlenbeck oscillator

We now consider the control of a 2D Ornstein-Uhlenbeck process [123], where each of the two state space variables is noise driven (see equation (4.64) below). As noted in [123], this system shares a few salient features of mechanical oscillators, e.g. the existence of a noisy spiral sink system or a noisy limit cycle. The model dynamics and the expression for the total cost are given below by equations (4.64) and (4.65) followed by the parameter values used for this problem.

$$d\boldsymbol{X}_t = \mathrm{A}\boldsymbol{X}_t dt + \mathrm{B} d\boldsymbol{B}_t \;\; ; \;\; t \in [0, 10] \tag{4.64}$$

$$\text{cost} : \boldsymbol{x}_{t_f}^T \mathrm{W} \boldsymbol{x}_{t_f} + \int_{t_0}^{t_f} \boldsymbol{x}_t^T \mathrm{Q} \boldsymbol{x}_t + \frac{1}{2} \boldsymbol{u}_t^T \mathrm{R} \boldsymbol{u}_t \tag{4.65}$$

$$\mathrm{A} = \begin{bmatrix} 0.47 & 1.25 \\ 0.75 & -0.53 \end{bmatrix} \mathrm{B} = \begin{bmatrix} 0.4861 & -0.1169 \\ -0.1169 & 0.3692 \end{bmatrix}$$
$$\mathrm{Q} = 50[\mathbb{I}]_{2\times 2} \;\; ; \;\; \mathrm{W} = 10[\mathbb{I}]_{2\times 2} \mathrm{R} = 0.1[\mathbb{I}]_{2\times 2} \;\; ; \;\; \mathrm{G} = 0.5[\mathbb{I}]_{2\times 2}$$

[1] Only those trajectories are chosen that stay within [-2,2]; 11 out of 50 trajectories go outside this interval for the FBSDE method.

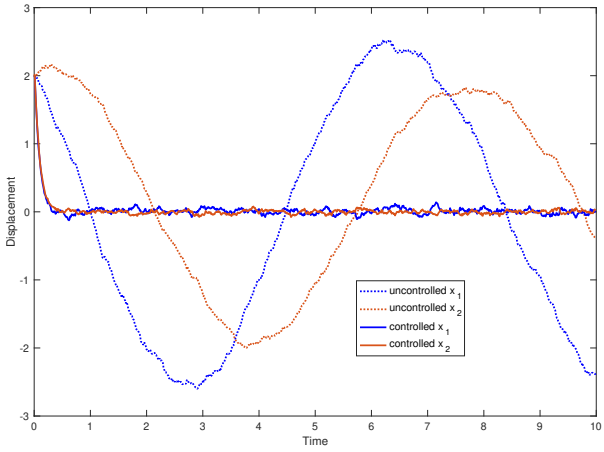


Figure 4.2: Comparison of controlled paths determined by the proposed method with the uncontrolled paths of a 2D Ornstein-Uhlenbeck process. Time is measured in seconds and displacement in meters.

$\mathbb{I}$ here denotes an identity matrix with dimension as indicated in the subscript. In Figure 4.2, we compare the results obtained via the proposed method and the uncontrolled dynamics. The efficacious working of our method is evident.

Lorenz oscillator

A recurrent theme for the problems considered hereafter is chaos. The study of control of chaos is of interest for two main reasons. The first is that, since these systems exhibit the 'butterfly effect', they form useful test problems to study the efficiency and robustness of any control methodology. Second, due to the same 'butterfly effect' which implies the presence of infinitely many unstable periodic orbits in the flow, the behaviour of chaotic systems may vary significantly for a small change in control. Thus it is possible, at least in principle, to dramatically reduce the total cost of control. In this section we study the control of the well known Lorenz 1963 [124] model. The parameter values chosen ensure that the behaviour of the uncontrolled system is chaotic. The equation of motion for this 3-dimensional model is given by

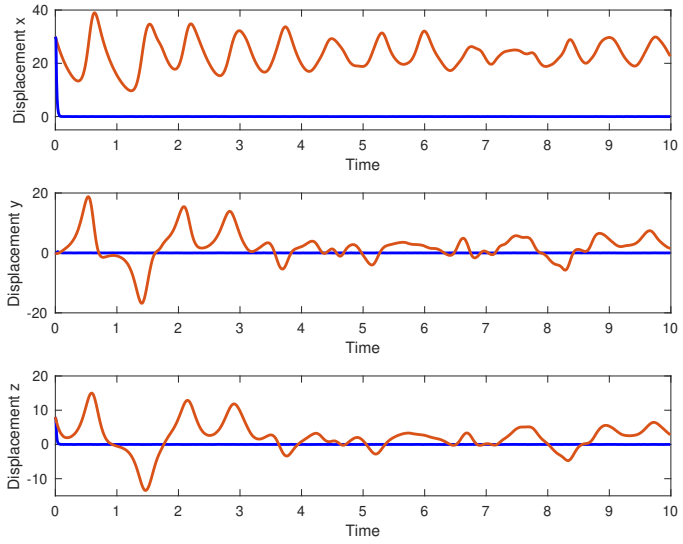


Figure 4.3: Comparison of controlled (blue) and uncontrolled (red) paths for the Lorenz 1963 model. Time is measured in seconds and displacement in meters.

$$d\boldsymbol{X}_t = \boldsymbol{f}(\boldsymbol{X}_t)dt + \Sigma d\boldsymbol{B}_t \tag{4.66}$$

$\boldsymbol{X}_t$ is the 3-dimensional vector $(x,y,z)^T$ co-ordinates and $\boldsymbol{f}$ is given by

$$\boldsymbol{f}(\boldsymbol{X}) = \begin{Bmatrix} \chi(y-x) \\ \rho x - y - xz \\ xy - bz \end{Bmatrix} \tag{4.67}$$

The cost is similar to the previous example, which is so designed that the controlled system should reach the origin in the phase space and thereafter stay there. The various parameter values used are

$$\chi = 10\ ;\ \rho = 28\ ;\ b = \frac{8}{3}\ ;\ \Sigma = 0.3[\mathbb{I}]_{3\times 3}\ ;\ \mathrm{Q} = 50[\mathbb{I}]_{3\times 3}\ ;$$
$$\mathrm{W} = 10[\mathbb{I}]_{3\times 3}\mathrm{R} = 0.1[\mathbb{I}]_{3\times 3}\ ;\ \mathrm{G} = 0.5[\mathbb{I}]_{3\times 3}$$

Figure 4.3 shows time history plots for all the three states corresponding to both controlled and uncontrolled cases. Given that the response regime is chaotic, our choice of the initial time to expand $g(\boldsymbol{x})$, which is presently t_0, is far from perfect.

This is, for instance, evidenced in a nearly immediate realization of the control objective that involves a high initial control cost. A more appropriate choice of $\bar{t}$ (to expand $g(\boldsymbol{x})$ about) is perhaps the first entry time of the uncontrolled trajectory within a ball of 'small' radius around the origin.

Hardening Duffing oscillator

Next we consider the control of a hardening Duffing oscillator. As mentioned in a previous subsection, this model may also behave chaotically. However, unlike the examples considered so far, the equation of motion for this model is not an SDE, but of the form shown in equation (4.41). Specifically, it is presently given by

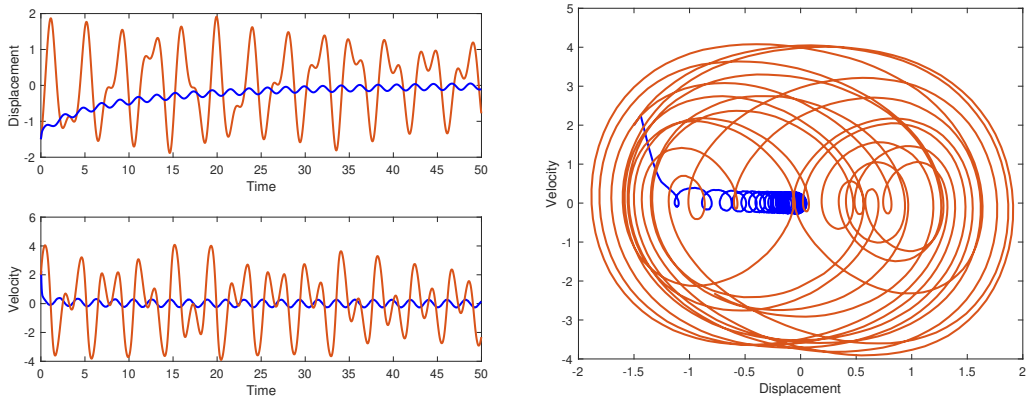
$$\ddot{\zeta} + c\dot{\zeta} + k\zeta + \alpha\zeta^3 = P\cos(2\omega t) + \sigma\dot{B} \tag{4.68}$$

In equation (4.68), c is the viscous damping coefficient, k, α are model stiffness parameters, P and ω are the forcing amplitude and frequency respectively and σ is the diffusion coefficient. $\dot{B}$ formally represents white noise, and in order to deal with it, the dynamics are re-written in the state space as follows

$$dX_1 = X_2 dt \tag{4.69}$$

$$dX_2 = (-cX_2 - kX_1 - \alpha X_1^3 + P\cos(2\omega t))dt + \sigma dB_t \tag{4.70}$$

where $X_1 = \zeta$ and $X_2 = \dot{\zeta}$ are respectively the displacement and velocity stochastic processes.The objective of control and hence the form of the cost are similar to the problems considered before. In this type of system, the control only acts on the equation for velocity which is an SDE. The expression for the total cost and the various parameters used for solving the stochastic optimal control problem with the proposed method are specified below. The results showing a comparison of controlled and uncontrolled trajectories are in Figure 4.4.



(a) **(b)**

Figure 4.4: Comparison of controlled (blue) and uncontrolled (red) paths for the hardening Duffing oscillator; (a) time history plots, (b) phase plots. Time is measured in seconds and displacement in meters.

$$\text{cost: } w_1 x_{1,t_f}^2 + w_2 x_{2,t_f}^2 + \int_{t_0}^{t_f} (q_1 x_{1,t}^2 + q_2 x_{2,t}^2 + \frac{1}{2} \mathrm{R} u_t^2) dt$$

$$P = 4 \; ; \; c = 0.1 \; ; \; k = 1 \; ; \; \alpha = 1 \; ; \; \omega = 3 \; ; \; \sigma = 0.1$$

$$q_1 = w_1 = 100 \; ; \; q_2 = w_2 = 30 \; ; \; \mathrm{R} = 1 \; ; \; \mathrm{G} = 0.5$$

$$\text{Initial conditions: } [x_{1,t_0}, x_{2,t_0}] = [1, 2]; \; t_0 = 0$$

Duffing-Holmes oscillator

As a final example, we consider the Duffing-Holmes oscillator [125, 126], where chaos occurs due to repeated intersections of the stable and unstable manifolds pertaining to the homoclinic orbit. For this particular example, we consider three cases in order to test the proposed method of control against different response regimes of the system depending on different initial conditions in the phase space. Perhaps, the most difficult to achieve is the control towards the unstable fixed point (the origin in the phase space), which is considered as the last case. The equation of motion along with the parameters used (that are common to all the cases) are given below. The cost and initial conditions specific to each case are described below in the appropriate places. As in the previous example, the control only acts on the second

equation which is an SDE in velocity.

$$\ddot{\zeta} + 2\pi\varepsilon_1\dot{\zeta} + 4\pi^2\varepsilon_2(\zeta^2 - 1)\zeta = 4\pi^2\varepsilon_3\cos(2\pi t) + \sigma\dot{B} \tag{4.71}$$

$$dX_1 = X_2 dt \tag{4.72}$$

$$dX_2 = (-2\pi\varepsilon_1 X_2 - 4\pi^2\varepsilon_2(X_1^2 - 1)X_1 + 4\pi^2\varepsilon_3\cos(2\pi t))dt + \sigma dB_t \tag{4.73}$$

$$\varepsilon_1 = 0.25 \; ; \; \varepsilon_2 = 0.5 \; ; \; \varepsilon_3 = 0.5 \; ; \; \sigma = 0.1$$

Case 1

In this case, for the initial condition chosen, the system behaviour is chaotic as evidenced by the presence of a strange attractor (or, equivalently, by the positivity of the highest Lyapunov exponent). The objective is to apply control so that it exhibits a 1-periodic orbit about one of the stable fixed points (in this case $[1,0]$) in the phase space. Note that the fixed point $[1,0]$, as is its counterpart at $[-1,0]$, is a stable sink when the oscillator is unforced. The cost chosen towards our control objective is given below along with the initial condition and the parameter values. The results for this case are shown in Figure 4.5.

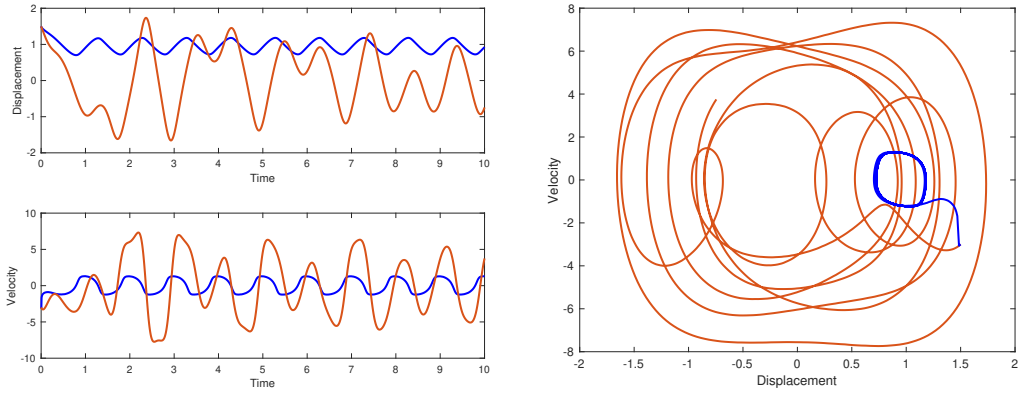
$$\text{Initial conditions: } [x_{1,t_0}, x_{2,t_0}] = [1.5, -3] \; ; \; t \in [t_0 = 0, t_f = 10]$$

$$\text{cost} : g_{t_f}(.) + \int_{t_0}^{t_f} (g_t(.) + \frac{1}{2}\mathrm{R}u_t^2)dt$$

$$g_t(.) = (w_1(x_{1,t} - 1)^2 + w_2 x_{2,t}^2 - r^2)^2$$

$$w_1 = 0.1 \; ; \; w_2 = 1 \; ; \; r = 0.5 \; ; \; \mathrm{R} = 0.1 \; ; \; \mathrm{G} = 0.5$$

It may be noted that, while controlling chaotic orbits, one can do better than Ito expanding $g(X_{t_f})$ about the initial time t_0. Specifically, the start time $\bar{t}$ to expand $g(X_{t_f})$ should be ideally chosen as the first time the chaotic orbit comes closest to the targeted (periodic) orbit in the phase space. This would substantially reduce the control cost.



(a) **(b)**

Figure 4.5: Comparison of controlled (blue) and uncontrolled (red) paths for the Duffing-Holmes oscillator; (a) time history plots, (b) phase plots. Time is measured in seconds and displacement in meters. For the parameters and initial condition chosen, the behaviour of the oscillator is chaotic. Time is measured in seconds and displacement in meters.

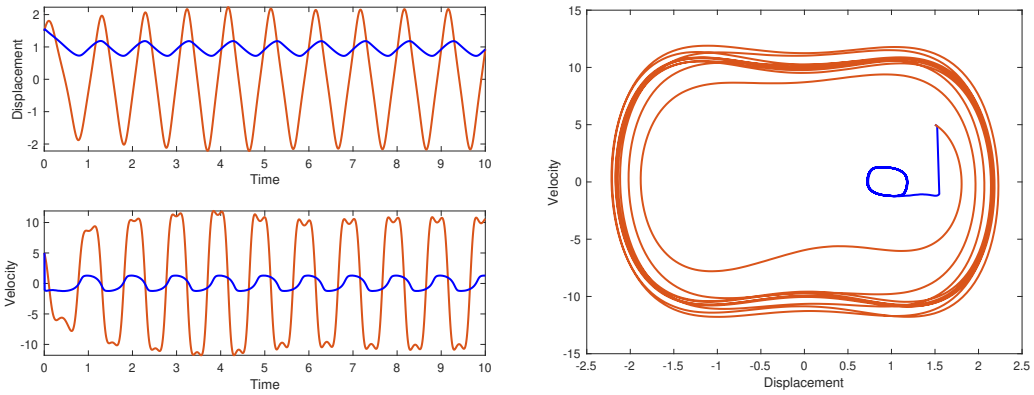
Case 2

This case is similar to Case 1, except that for the initial condition chosen, the system behaviour is not chaotic, but in the form of a large periodic orbit that encapsulates the homoclinic orbit, and hence all three fixed points for the unforced oscillator. The control is designed so as to exhibit a periodic behaviour with a much smaller amplitude. This, among others, demonstrates that the proposed method for control is robust irrespective of whether the system behaviour is chaotic, periodic or quasi-periodic. The results for this case are depicted in Figure 4.6.

$$\text{Initial conditions: } [x_{1,t_0}, x_{2,t_0}] = [1.5, 5] \;\; ; \;\; t \in [t_0 = 0, t_f = 10]$$
$$\text{cost} : g_{t_f}(.) + \int_{t_0}^{t_f} (g_t(.) + \frac{1}{2}\mathrm{R}u_t^2)dt$$
$$g_t(.) = (w_1(x_{1,t} - 1)^2 + w_2 x_{2,t}^2 - r^2)^2$$
$$w_1 = 0.1 \; ; \; w_2 = 1 \; ; \; r = 0.5 \; ; \; \mathrm{R} = 0.1 \; ; \; \mathrm{G} = 0.5$$

Case 3

Finally, we control the Duffing-Holmes oscillator to stay at the unstable fixed point i.e. $[0,0]$ in the phase space. The initial point is chosen close to the origin in the



(a) **(b)**

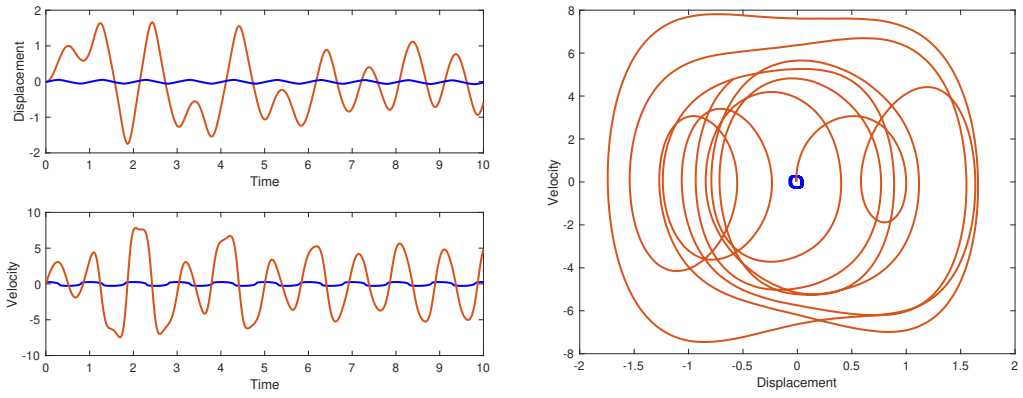
Figure 4.6: Comparison of controlled (blue) and uncontrolled (red) paths for the Duffing-Holmes oscillator; (a) time history plots, (b) phase plots. Time is measured in seconds and displacement in meters. For the parameters and initial condition chosen, the behaviour of the oscillator is non-chaotic. Time is measured in seconds and displacement in meters.

phase space; the uncontrolled trajectories are naturally repelled from the origin in this case as can be seen in Figure 4.7(a) and hence they are not easy to control. The expression for cost, and the various parameters used are given below. Results for the controlled dynamics are shown in Figure 4.7.

$$\text{Initial conditions: } [x_{1,t_0}, x_{2,t_0}] = [0.01, 0.01] \;;\; t \in [t_0 = 0, t_f = 10]$$
$$\text{cost} : g_{t_f}(.) + \int_{t_0}^{t_f} (g_t(.) + \frac{1}{2}\mathrm{R}u_t^2)dt$$
$$g_t(.) = (w_1(x_{1,t})^2 + w_2 x_{2,t}^2)^2$$
$$w_1 = 1 \;;\; w_2 = 10 \;;\; \mathrm{R} = 0.1 \;;\; \mathrm{G} = 0.5$$

4.5 Concluding remarks

Unlike the well known HJB equation supplied with a terminal condition to characterize a stochastic control problem, we, in this chapter, have derived a PDE with an initial condition to represent the same control problem. Our proposal to solve the PDE involves a pair of SDEs both integrated forward in time within a Monte Carlo setting plus a functional approximation scheme over every time step. This may be



(a) **(b)**

Figure 4.7: Comparison of controlled (blue) and uncontrolled (red) paths for the Duffing-Holmes oscillator; (a) time history plots, (b) phase plots. Time is measured in seconds and displacement in meters. Time is measured in seconds and displacement in meters.

contrasted with the far more tedious route of solving the HJB PDE via a pair of FBSDEs. In the proposed makeover of the stochastic control problem, while the total cost to minimize and the control objectives remain the same, the terminal cost is smeared over the time interval of interest by Ito's formula to arrive at an initial cost. This means that the pathwise solutions in our Monte Carlo scheme could be different from those in the standard FBSDE approach. We arrived at the same PDE using two complementary approaches, thus confirming that the difference between the proposed method and the one via the HJB route lies entirely in the Ito expansion of the terminal cost and not on whether the cost-to-go or cost-to-reach is being minimized. We also give a brief derivation of our approach for the special case of mechanical oscillators, which constitutes an application of primary interest in this work.

Although there are no serious restrictions on the process dynamics and the running cost involving the system states, the control term in the system dynamics is assumed to be linear and the cost associated with control quadratic; this is also known as $\mathscr{L}^2$ control. The scheme may be adapted for $\mathscr{L}^1$ control without much difficulty, although we do not elaborate on those scenarios in this work. Also, we only considered problems with a finite time horizon, i.e. when the terminal time is fixed beforehand. Our proposal however admits ready extensions to problems

involving a stopping time, or an exit time since the SDEs in the method evolve only forward in time. The ease of implementation, computational expedience, and scalability are a few of the major advantages of our scheme vis-á-vis the ones through forward-backward stochastic dynamics.

It is known that the SDE-based route to solve PDEs, such as the HJB equation, may not efficaciously explore the spatial domain over which solutions may be sought. As with the scheme based on FBSDEs, it is possible to engineer the drift terms, Girsanov change of measures to wit, towards exploring specific spatial regions of interest. Several other interesting directions for future exploitation of the proposed scheme could be suggested. Application to stochastic filtering is one such possibility, where the control terms could be designed for variance reduced estimates of states. Parameter estimation within a stochastic filter is yet another possibility. Here, for instance, the parameter dynamics driven only by Brownian noise might be enhanced with an additional drift or control to precipitate faster convergence. Design of a control strategy within a Riemannian geometric setting, with the metric derived through the cost, is also an exciting direction for future research.

The next chapter is about parameter estimation via data assimilation. At first glance, the methodology of data assimilation, in particular the ensemble Kalman filter (for either states alone or states and parameters combined), may seem to bear a remarkable similarity with the solution for the SOC problem considered here. More specifically, an additional term is derived that is added to the process SDE like the Gu term in equation 4.23. Indeed, as mentioned in Chapter 1, these two problems may be looked upon as two sides of the same coin. The key difference in these two problems lies in their objectives. In SOC, the objective is to reach a desired terminal state whilst minimizing a certain total cost and the additional drift derived is a force that needs to be applied to the system in order to achieve that objective. In data assimilation on the other hand, the objective is to determine the true values of states (and parameters) of a dynamical system using the available measurements of a system evolution that has already occured. In this case, therefore, the additional term is like a pseudo-force that drives the mean of the innovation vector (defined

in the next chapter) to zero. Although not explicit at first glance, it is also worth appreciating the connection between the parameter dynamics proposed in the next chapter and the optimization and Markov chain Monte Carlo methods in Chapters 2 and 3.

Chapter 5

Efficient parameter estimation via data assimilation: a modified parameter dynamics

While I have undertaken the theoretical work for this chapter and the numerical work in Section 5.5.1, the numerical examples in Sections 5.5.2 and 5.5.3 have also been contributed to by some of my collaborators. The programming work for Section 5.5.2 has been carried out together with Marcus Lehmann, Dr.-Ing., Ariane Group GmbH, while the text in that section is produced entirely by him. Except for some debugging which was undertaken together, the programming work for Section 5.5.3 has been mainly carried out by Dr. Devaraj Gopinathan, Centre for Advanced Research Computing, UCL, who has also co-drafted the text therein.

5.1 Introduction

Given a system model and a partial set of noisy observations on the system, a data assimilation (DA) method aims at estimating the states of the dynamical system whilst acknowledging the modelling and observation uncertainties. A stochastic filtering method, which adopts a Bayesian probabilistic framework to obtain the estimates [127], typically starts with a prior (or proposal), informed by the system model, which is then updated by observations to obtain the posterior (prior times the likelihood) and the related estimates of state thereof as new observations become

available. These estimates are the mean of the system's states or parameters conditioned on the available observations. They cannot generally be found in closed form. This is due to the nonlinearities in the process and/or observations models as well as possible non-Gaussianity in describing the uncertainties. Accordingly, ensemble methods or those based on Monte Carlo simulations are mostly adopted wherein posterior distributions are approximated using sampling. Prominent among the Monte Carlo based DA methods are particle filters, of which the bootstrap filter is perhaps the oldest [128]. Here, as in most other stochastic filters, the system dyncamics is recursively predicted and updated. The prediction step being performed by integrating the system dynamics to obtain an ensemble of particles thus resulting in a sampling approximation to the prior. The current observation is then used to compute the likelihood function and thus assign weights to the predicted particles, which forms the update step. The sampling approximation to the posterior is then the ensemble of weighted particles which may be resampled to arrive at particles with equal weights.

The auxiliary particle filter [129] improves upon the complexity of the bootstrap filter algorithm by considering the posterior in the product space of states/parameters at the last and current time instants. Quasi-Monte Carlo filters [130, 131] improve the convergence of the Monte Carlo simulations that underlie the filtering scheme by replacing random draws by more regular samples. A major numerical issue with most particle filters is that of particle degeneracy which is reflected in all but one weight reducing to zero as time-recursions progress [132]. We refer to [133] for a recent review on particle filters.

Notable among the ensemble methods are the ensemble Kalman filters with the popular acronym 'EnKF' [134, 135] which employ Gaussian assumptions for both the prior and the likelihood. Since these are ensemble makeovers of the well known Kalman filter, the update strategy for the predicted ensemble is not based on weights, but an additive correction term. The correction term is typically of the form of a gain matrix multiplying the innovation vector, which measures the error in the predicted observation. Accordingly, the particle degeneracy problem is not encoun-

tered in this class of methods. Following the original work in [134], several variants of this class of filters have been developed, e.g. the ensemble transform Kalman filter [136], the ensemble adjustment Kalman filter [137], the ensemble square root filter [138, 139], the maximum likelihood ensemble filter [140], the error subspace transform Kalman filter [141] and several others. These variants of the EnKF mainly differ in the details of the update step. These filters clearly cannot account for possible non-Gaussianity in the model or observation uncertainty. This shortcoming is overcome in the Kushner-Stratonovich filter [142]. Though computationally more expensive, it can account for non-Gaussianity via an ensemble-based implementation of the nonlinear filtering equation – the Kushner-Stratonovich equation [26].

We now turn to the focus of this work, wherein our interest is in estimating parameters of a given dynamical system within the framework of data assimilation. Without any loss of generality, which will be made more explicit later on, we work with the EnKF. When the EnKF is used for estimating parameters, the conventional approach is to append the vector of parameters to the original state vector [143]. The state vector is made to evolve as usual according to the process dynamics equation, with the exception that now the parameter vector is also considered to evolve with time. Barring a few exceptions viz. [144, 145] the evolution of the parameter vector is implemented using Brownian motion. This combined evolution of state and parameters forms the prediction step. The update step is performed as usual, except the dimension of Kalman matrix update step is now different (larger) than it would have been had we been interested in only state estimation. Overall, the parameter prediction, which is taken as Brownian motion in the prediction step, is improved at the update step, see [146] for an example where the ensemble Kalman filter is used for combined state-parameter estimation. Although this works well in many practical applications, the Brownian motion proposal for the prediction of parameters is quite ad hoc and may cause several issues in general. For instance, if the process is sensitive to some or all of the desired parameters, Brownian motion proposal tends to perform poorly. Also, this proposal performs quite poorly when the availability of data is limited. Limited data may either mean the frequency of data for a given

time interval is lower or that the total time over which data is collected is itself quite small. Also, it may become increasingly difficult to achieve convergence of parameters as the problem dimension increases. This type of proposal typically exhibits a large sampling variance as well, i.e. repeated simulations of the same problem may lead to the convergence of parameters to quite different values.

It is thus clear that Brownian motion alone is inadequate for parameter dynamics. In view of this, a novel scheme for estimating parameters of a dynamical system within the framework of data assimilation is proposed in this chapter. Specifically, this is achieved by constructing a Langevin dynamics for the evolution of parameters at the prediction step. The main element required in the construction of Langevin dynamics is an energy-like quantity which is proposed as a convex function of the innovation term in filtering. As a quick aside, this is not unlike the approach adopted in [8] for the design of a stochastic Hamiltonian preserving integration scheme. In the context of filtering, at any given time, the innovation term may be defined as the difference between the measurement and its predicted value based on the available information about the predicted states and the measurement model dynamics. The gradient of this energy-like quantity with respect to the desired parameters forms the drift of the Langevin diffusion equation and gives a direction to the evolution of the parameters facilitating faster search. This approach of parameter evolution is in contrast with the conventional one mentioned in the previous paragraph, wherein the parameter dynamics is simply Brownian motion and hence has no directional information. We name our method PEDAL for Parameter Estimation with Data Assimilation using Langevin Dynamics. Moreover, PEDAL affects only the prediction step of filtering, and is thus suitable to be applied within both the gain-based filtering methods like the EnKF and the weight-based ones like the Bootstrap filter. In addition to being generic enough to be applied within any type of filtering approach, PEDAL is also easy to implement. The superior performance in speed and accuracy of convergence of the proposed approach, when compared to the conventional approach, is demonstrated through a few numerical examples. Through further examples, where the frequency of data is lower, we show

that the conventional approach is unable to match the performance of the proposed one even with an ensemble size of more than 100 times for the Lorenz oscillator. We also demonstrate the robustness of the proposed approach through repeated simulations.

The rest of this chapter is organised as follows. For the sake of completeness, Section 5.2 describes the problem of state estimation in filtering along with a pseudo-code for state estimation via the EnKF. Building upon Section 5.2, 5.3 describes the problem of combined state-parameter estimation; again a pseudo-code is included for the sake of clarity. The methodology is discussed in detail in Section 5.4, which is followed by numerical examples and conclusions in sections 5.5 and 5.6 respectively.

5.2 State estimation via ensemble Kalman filter

Consider a filtering problem with the following equations for process and measurement dynamics.

$$dX_t = f(X_t, \theta)dt + \sigma_x dB_t \tag{5.1}$$

$$Y_t = h(X_t) + \sigma_z dW_t \tag{5.2}$$

Here X_t is the vector of model states, θ the vector model parameter and Y_t, the measurement vector at time t. $f(.)$ and $h(.)$ are process and measurement dynamics, σ and σ_z, the process and measurement noise intensities, while dB_t and dW_t are independent Brownian noise increments. For simplicity, and without any loss of generality, we assume that observations are available at every time increment. The objective of filtering is to incorporate these observations within the available process model to estimate the optimal solution whilst acknowledging the uncertainties in the process dynamics as well as observations. As mentioned in the introduction, any DA scheme typically consists of two steps - the first being prediction, where the state variable X_t is passed through the process model to obtain the predicted state at the next time instant $\tilde{X}_{t+1}$. In the second step, the latest measurement Z_{t+1} is assimilated to obtain the updated or filtered state estimate $\hat{X}_{t+1}$. For clarity of exposition, a pseudo-code for the EnKF for state estimation is provided in Algorithm 4.

Algorithm 4: Pseudo-code of EnKF for state estimation

Let the states and measurements be d_X and d_Z dimensional vectors respectively
Discretize time ; $0 = t_0 < t_1 < ... < t_N = T$ such that $t_k - t_{k-1} = \Delta t \ \ \forall\, k \in [1,N]$
Data: d_Z-dimensional measurement vectors $\{Z\}_{k=1}^{N}$ at the discretized time steps
Initialize the ensemble (of size M) of states $\{\hat{X}^j\}_{k=0} \ \ \forall j \in [1,M]$ at time t_0
for $k = 0 : N-1$ **do**
  Predict
  for $j = 1 : M$ **do**
    $\tilde{X}_{k+1}^{j} = \hat{X}_k^j + f(\hat{X}_k^j, \theta)\Delta t + \sigma_x \Delta B_k \ \ \forall j \in [1,M] \ \ ; \ \ \Delta B_k \sim \mathcal{N}(0, \Delta t)$
  end
  Update
  Let $\tilde{X}_{k+1}$ and $\overline{\tilde{X}}_{k+1}$ represent the state ensemble and its mean respectively
  $P_k = \frac{1}{\sqrt{N-1}} \left[\tilde{X}_{k+1} - \overline{\tilde{X}}_{k+1}\right]_{d_X \times M}$

  Similarly, let $h(\tilde{X}_{k+1})$ and $\overline{h(\tilde{X}_{k+1})}$ represent the ensemble of predicted measurements and its mean respectively
  $Y_k = \frac{1}{\sqrt{N-1}} \left[h(\tilde{X}_{k+1}) - \overline{h(\tilde{X}_{k+1})}\right]_{d_Z \times M}$
  Determine the Kalman gain matrix K_k as follows
  $K_k = P_k Y_k^T [Y_k Y_k^T + \sigma_z \sigma_z^T]^{-1}$ where $(.)^T$ represents the matrix transpose
  for $j = 1 : M$ **do**
    Determine the updated ensemble of states $\hat{X}_{k+1}^j = \tilde{X}_{k+1}^j + K_k \left\{ Z_{k+1} - h\left(\tilde{X}_{k+1}^j\right) \right\} \ \ \forall j \in [1,M]$
  end
end
return $\hat{X}_{k+1}^j \ \ \forall j \in [1,M]$

5.3 Combined state-parameter estimation via ensemble Kalman filter

The combined state-parameter estimation proceeds very similarly to the state estimation. The only difference is now that the parameter is unknown, it is considered as a pseudo-state which also evolves with time, thus modifying the equations 5.1 and 5.2 to the following

$$dX_t = f(X_t, \theta_t)dt + \sigma_x dB_t \tag{5.3}$$

$$d\theta_t = \sigma_\theta dV_t \tag{5.4}$$

$$Y_t = h(X_t) + \sigma_z dW_t \tag{5.5}$$

where, like dB_t and dW_T, dV_t is also an independent Brownian noise increment. Thus, in addition to the equation for evolution of states, there is now also an equation for the evolution of the unknown parameter θ. σ_θ is the noise intensity for the evolution of parameters. The EnKF is now applied to the combined state-parameter vector, unlike only the state vector in the previous section. For clarity, the modified pseudo-code for is presented below. Note the only differences with the Algorithm 4 are that the prediction happens for both states and parameters and the Kalman gain matrix is of a larger size.

Algorithm 5: Pseudo-code of EnKF for combined state-parameter estimation: Brownian motion for parameters

Let the states, parameters and measurements be d_X, d_θ and d_Z dimensional vectors respectively

Discretize time ; $0 = t_0 < t_1 < ... < t_N = T$ such that $t_k - t_{k-1} = \Delta t \ \ \forall\, k \in [1,N]$

Data: d_Z-dimensional measurement vectors $\{Z\}_{k=1}^{N}$ at the discretized time steps

Initialize the ensemble (of size M) of states $\{\hat{X}^j\}_{k=0}$ and parameters $\{\hat{\theta}^j\}_{k=0}\ \forall\, j \in [1,M]$ at time t_0

for $k = 0 : N-1$ **do**

  Predict

  for $j = 1 : M$ **do**

$$\tilde{X}^j_{k+1} = \hat{X}^j_k + f(\hat{X}^j_k, \hat{\theta}^j_k)\Delta t + \sigma_x \Delta B_k \ \ \forall\, j \in [1,M] \ ; \ \ \Delta B_k \sim \mathcal{N}(0,\Delta t)$$

$$\tilde{\theta}^j_{k+1} = \hat{\theta}^j_k + \sigma_\theta \Delta V_k \ \ \forall\, j \in [1,M] \ ; \ \ \Delta V_k \sim \mathcal{N}(0,\Delta t)$$

  end

  Update

$$P_k = \frac{1}{\sqrt{N-1}} \left[\begin{Bmatrix} \tilde{X}_{k+1} \\ \tilde{\theta}_{k+1} \end{Bmatrix} - \overline{\begin{Bmatrix} \tilde{X}_{k+1} \\ \tilde{\theta}_{k+1} \end{Bmatrix}} \right]_{(d_X+d_\theta)\times M}$$

$$Y_k = \frac{1}{\sqrt{N-1}} \left[h(\tilde{X}_{k+1}) - \overline{h(\tilde{X}_{k+1})} \right]_{d_Z \times M}$$

  Determine the Kalman gain matrix K_k as follows

$$K_k = P_k Y_k^T [Y_k Y_k^T + \sigma_z \sigma_z^T]^{-1}$$

  for $j = 1 : M$ **do**

    Determine the updated ensemble of states and parameters

$$\begin{Bmatrix} \hat{X}^j_{k+1} \\ \hat{\theta}^j_{k+1} \end{Bmatrix} = \begin{Bmatrix} \tilde{X}^j_{k+1} \\ \tilde{\theta}^j_{k+1} \end{Bmatrix} + K_k \left\{ Z_{k+1} - h\left(\tilde{X}^j_{k+1} \right) \right\} \ \ \forall\, j \in [1,M]$$

  end

end

return $\hat{X}^j_{k+1}, \hat{\theta}^j_{k+1} \ \ \forall\, j \in [1,M]$

5.4 Proposed methodology

PEDAL consists of a modified parameter dynamics that replaces equation 5.4, leaving the rest of the filtering procedure exactly the same as in Section 5.3. As mentioned earlier, the modified dynamics is a Langevin diffusion equation, specifically, we use the overdamped Langevin diffusion. The main element required in constructing this diffusion is an energy-like quantity. Let the energy-like term at any time t_k be defined as E_k for now, it will be discussed a bit later in this Section. Then PEDAL (after an integration step as per the Euler-Maruyama method) is given as follows:

$$\theta_{k+1} = \theta_k - \beta_k \nabla_\theta E_k \Delta t + \sqrt{2\beta_k}\Delta V_k \tag{5.6}$$

In the equation above, $\nabla_\theta E_k$ is the gradient of the energy-like quantity with respect to the parameter vector, while Δt and ΔV_k are increments in time and Brownian motion respectively. β_k is an annealing parameter such that $\beta_k > 0 \; \forall k$, $\beta_{k+1} \leq \beta_k$. This is a temperature-like quantity; a higher value facilitates more exploration and a lower one exploitation. In the initial steps, when little data has been assimilated, the β value is kept high to allow more exploration and gradually reduced when more and more information becomes available with time. Thus $-\nabla_\theta E_k$ is the direction of descent and $\beta_k \Delta t$ may be considered analogous to the step length in a deterministic gradient descent algorithm. The drift term of the SDE (5.6) is proportional to β_k while the diffusion term is proportional to $\sqrt{2\beta_k}$ in accordance with the fluctuation-dissipation theorem [147].

Let us now turn our attention to the energy-like quantity E_k in equation 5.6. Since it is desired that E_k mimic an energy function, certain considerations must be kept in mind. First, it must be a non-negative function of the innovation process. Second, it should preferably be a convex function, with a unique minimum at the desired solution. Keeping these points in mind, we propose the following expression for a pseudo-energy as a function of the innovation process at any give time t_k.

$$E_k = \exp(\lambda (Z_k - h(\tilde{X}_k))^T (Z_k - h(\tilde{X}_k))) - 1 \;\; ; \;\; \lambda > 0 \tag{5.7}$$

Here λ is a positive hyperparameter, Z_k is the measurement and $\tilde{X}_k$, the predicted state vector at time t_k. Intuitively, the exponential function does the job of exaggerating the influence of the error or the innovation term on the energy, thus helping in a faster search. However, sometimes this influence can cause numerical issues, particularly in the initial time steps of filtering when the innovation is large, leading to large steps accompanied by large noise (since β_k is also relatively large in the initial time steps) not necessarily in the right directions. Hence, λ is introduced to soften this influence. Its typical value is much smaller than 1. Its actual value depends upon the nature of the process dynamics. For example, if the process is highly sensitive to the change in the values of one or more parameters, a larger value may be more suitable. It is also possible to choose a vector λ, of the same dimension as Z_k, in problems that are relatively more stiff with respect to some parameters than others.

Once the form of E_k is decided, we can now proceed to determine $\nabla_\theta E_k$. In this work, we only work with equation 5.7. Hence, its gradient, upon application of a simple chain rule, may be written in component form as follows.

$$\{\nabla_\theta E_k\}^i = \frac{\partial E_k}{\partial \theta^i} \tag{5.8}$$

$$= -2\lambda E_k (Z_k^p - h^p(\tilde{X}_k)) \frac{\partial h^p(\tilde{X}_k)}{\partial \tilde{X}_k^j} \frac{\partial \tilde{X}_k^j}{\partial \theta^i}\bigg|_{\theta=\theta_k} \tag{5.9}$$

In the equation above, the subscript k represents the time index. A superscript $(.)^m$ is used to represent the m^{th} component of the vector quantity under consideration. Also, the Einstein summation convention has been made use of, meaning a sum is implied over any repeated indices (i.e. 'p' and 'j' in equation 5.9) in the superscript. Looking at individual terms appearing in equation 5.9, E_k is determined as per equation 5.7, $(Z_k^p - h^p(\tilde{X}_k))$ is the p^{th} component of the innovation vector, $\frac{\partial h^p(\tilde{X}_k)}{\partial \tilde{X}_k^j}$ is the partial derivative of p^{th} the equation in the measurement model with respect to the j^{th} component of the state vector and $\frac{\partial \tilde{X}_k^j}{\partial \theta^i}$ is the partial derivative of the j^{th} component of the state vector with respect to the i^{th} component of the parameter vector. While the other quantities are straightforward, in order to determine the last partial

derivative, we make use of the implicit Euler-Maruyama integration of the process dynamics as follows.

$$X_k = X_{k-1} + f(X_k, \theta)\Delta t + \sigma_x \Delta B_k \tag{5.10}$$

Based on the equation above, the last term in equation 5.9 can now be determined as

$$\frac{\partial \tilde{X}_k^j}{\partial \theta^i} = \left.\frac{\partial f^j(\tilde{X}_k, \theta)}{\partial \theta^i}\right|_{\theta=\theta_k} \tag{5.11}$$

The explicit Euler-Maruyama method of integration may also have been used instead, we opt to choose the implicit method for convenience. More specifically, we may have had to keep track of states at two time steps in the past with the explicit method instead of just one with the implicit one. Note that θ in the numerator in equation 5.11 represents the vector of parameters.

5.5 Numerical experiments

We demonstrate the performance of PEDAL with the help of three examples. The first is the Lorenz 63 model [124], which is a well-known benchmark chaotic oscillator. The next is a standard material model of linear elastic behaviour. The last example is of tsunami propagation. This does not naturally fall in the class of state-parameter estimation, as will be made clearer later, since the parameters we are trying to infer therein are those of the initial tsunami swell. These parameters basically define the initial state for the tsunami, and in the data-assimilation set-up, are simply states. We show in Section 5.5.3 how we use a workaround and treat the parameters for the initial condition as parameters in the system dynamics. In all three examples, we compare the results of the filtered states and parameters with the EnKF between the conventional and proposed approaches.

5.5.1 Lorenz 1963 oscillator

This is a 3-dimensional model, with three parameters σ, r and b. The process model evolution, i.e. $f(.)$ in equation (5.3), is given by the following set of coupled ordi-

Algorithm 6: Pseudo-code of ensemble Kalman filter for combined state-parameter estimation: proposed method

Let the states, parameters and measurements be d_X, d_θ and d_Z dimensional vectors respectively

Discretize time ; $0 = t_0 < t_1 < ... < t_N = T$ such that $t_k - t_{k-1} = \Delta t \ \ \forall\, k \in [1,N]$

Data: d_Z-dimensional measurement vectors $\{Z\}_{k=1}^{N}$ at the discretized time steps

Initialize the ensemble of states (of size M) $\{\hat{X}^j\}_{k=0}$ and parameters $\{\hat{\theta}^j\}_{k=0} \ \ \forall j \in [1,M]$ at time t_0

for $k = 0 : N-1$ **do**

  Predict

  for $j = 1 : M$ **do**

    $\tilde{X}_{k+1}^j = \hat{X}_k^j + f(\hat{X}_k^j, \hat{\theta}_k^j)\Delta t + \sigma \Delta B_k \ \ \forall j \in [1,M] \ ; \ \Delta B_k \sim \mathcal{N}(0, \Delta t)$

    $\tilde{\theta}_{k+1}^j = \hat{\theta}_k^j - \beta_k \nabla_\theta E_k(\hat{\theta}_k^j)\Delta t + \sqrt{2\beta_k}\Delta V_k \ \ \forall j \in [1,M] \ ; \ \Delta V_k \sim \mathcal{N}(0, \Delta t)$

    where E_k is defined in equation (5.7) and $\nabla_\theta E_k(\hat{\theta}_k^j)$ is determined using equations (5.9) and (5.11)

  end

  Update

  $P_k = \frac{1}{\sqrt{N-1}} \left[\begin{Bmatrix} \tilde{X}_{k+1} \\ \tilde{\theta}_{k+1} \end{Bmatrix} - \overline{\begin{Bmatrix} \tilde{X}_{k+1} \\ \tilde{\theta}_{k+1} \end{Bmatrix}} \right]_{(d_X + d_\theta) \times M}$

  $Y_k = \frac{1}{\sqrt{N-1}} \left[h(\tilde{X}_{k+1}) - \overline{h(\tilde{X}_{k+1})} \right]_{d_Z \times M}$

  Determine the Kalman gain matrix K_k as follows

  $K_k = P_k Y_k^T [Y_k Y_k^T + \sigma_z \sigma_z^T]^{-1}$

  for $j = 1 : M$ **do**

    Determine the updated ensemble of states and parameters

    $$\begin{Bmatrix} \hat{X}_{k+1}^j \\ \hat{\theta}_{k+1}^j \end{Bmatrix} = \begin{Bmatrix} \tilde{X}_{k+1}^j \\ \tilde{\theta}_{k+1}^j \end{Bmatrix} + K_k \left\{ Z_{k+1} - h\left(\tilde{X}_{k+1}^j \right) \right\} \ \ \forall j \in [1,M]$$

  end

end

return $\hat{X}_{k+1}^j, \hat{\theta}_{k+1}^j \ \ \forall j \in [1,M]$

nary differential equations (ODEs).

$$\begin{aligned} \frac{dx}{dt} &= \sigma(y-x) \\ \frac{dy}{dt} &= rx - y - xz \\ \frac{dz}{dt} &= xy - bz \end{aligned}$$

The true parameter values are taken as, $\sigma = 10, r = 28, b = 8/3$ and the initial condition for the state used for generating toy measurements is $x = 8, y = 0, z = 30$. Parameter noise intensity for the Brownian motion case is taken as $\sigma_\theta = 0.1$, pro-

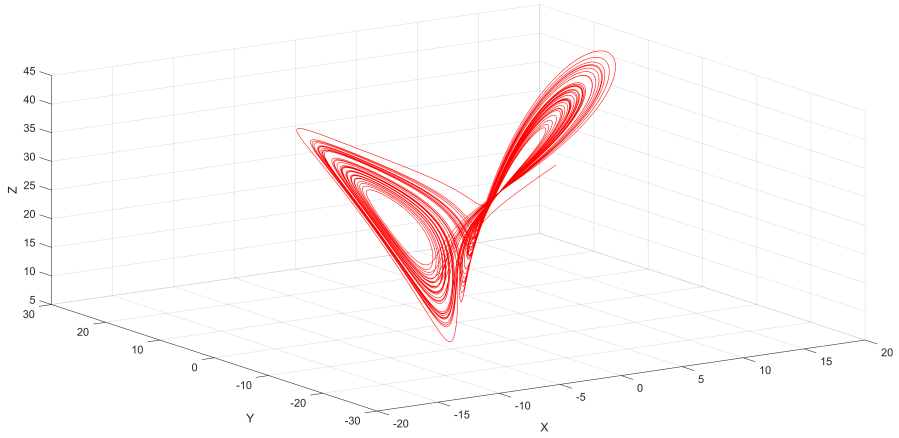


Figure 5.1: Typical phase plot of the Lorenz 63 model

cess noise intensity $\sigma_x = 0.001$ and the measurement noise intensity is taken as $\sigma_z = 2$. In general, an explicit Euler-Maruyama method of integrating SDEs works well for most practical applications. However, since this is a chaotic system, the stochastic Heun method of integration is used for integrating the process dynamics in the prediction step of the EnKF. This may be considered as the stochastic counterpart of the second-order Runge-Kutta method of integrating ODEs. Figure 5.1 shows the typical phase plot of the Lorenz 63 model. It is interesting to note that a linear problem with known parameters may become non-linear when the parameters are unknown, as is true for the Lorenz oscillator. In the context of filtering, particularly the EnKF, this introduces a complexity that is more than just an increase in the problem dimension.

We perform three types of experiments to test the quality of results. The first one is a simple comparison of the conventional and proposed methods when data is assimilated at every time step. The results are shown in Figure 5.2. Of specific interest are the plots for the parameters, i.e. subfigures (a), (c) and (e). Each of these subfigures contains results of 4 different instances of the EnKF. The solid lines in green and pink are the results with the Brownian (original) and Langevin (modified) parameter dynamics respectively, each for an ensemble size of 10. As is clear from the subfigures, PEDAL performs much better, particularly for the parameter σ. The dashed lines in orange and blue are results with Brownian motion for an ensemble

size of 100 and 1000 respectively. It may be seen that even with an ensemble size of 1000 (dashed blue line), the Brownian motion cannot outperform the proposed method with an ensemble size of just 10 (solid line in pink).

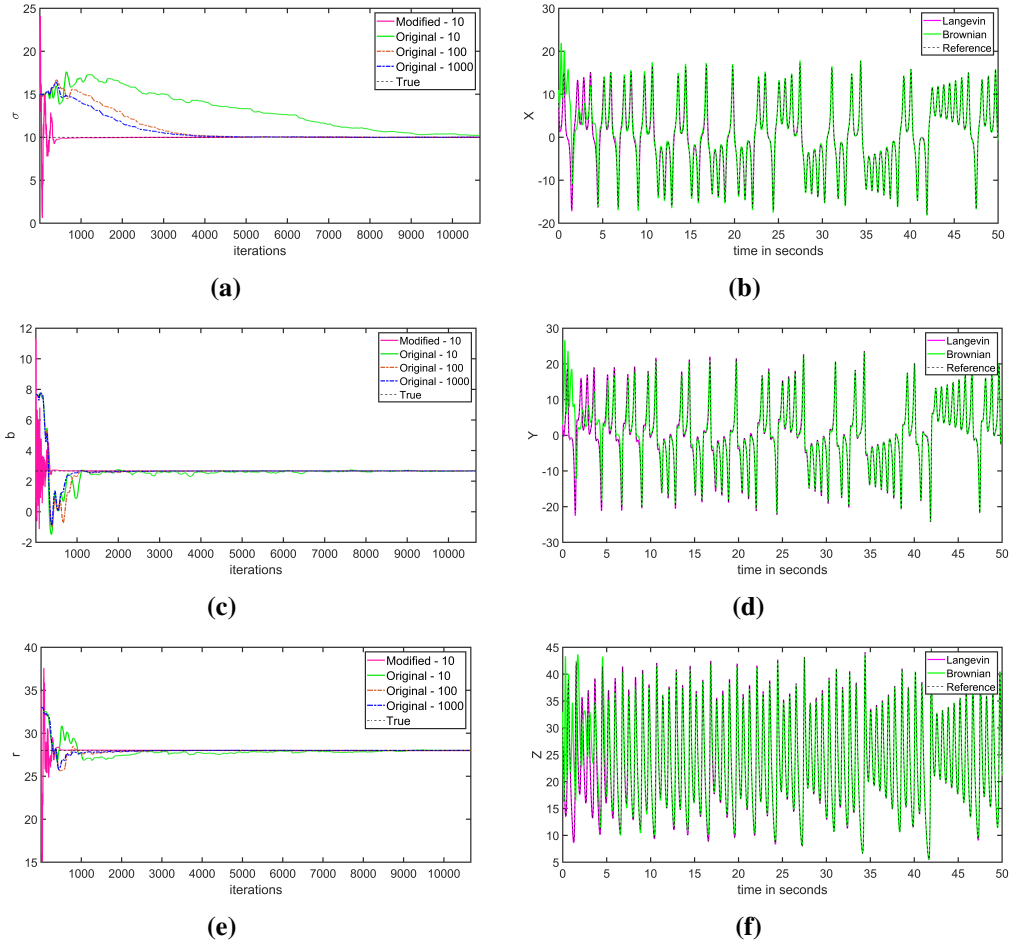


Figure 5.2: Lorenz 63 model: Comparison of filtered states and parameters between the two parameter dynamics, i.e. Brownian (original) and Langevin (modified). Figures (a), (c) and (e) show the convergence of parameters. The numbers accompanying the labels represent the ensemble size used in the EnKF. Figures (b), (d) and (f) show the convergence of the three states for an ensemble size of 10. Data is assimilated at every time step.

The second set of experiments is the same as the first one, except now the data is assimilated only every tenth time step. That is, the availability of data to the filtering algorithm is 10 times lower since the total time of simulation is left unchanged. The objective is twofold; one, to demonstrate that the requirement of data with the proposed method is less, thus reducing the computational cost of filtering. Two, the available data (over which one may have little control) is made better use of with

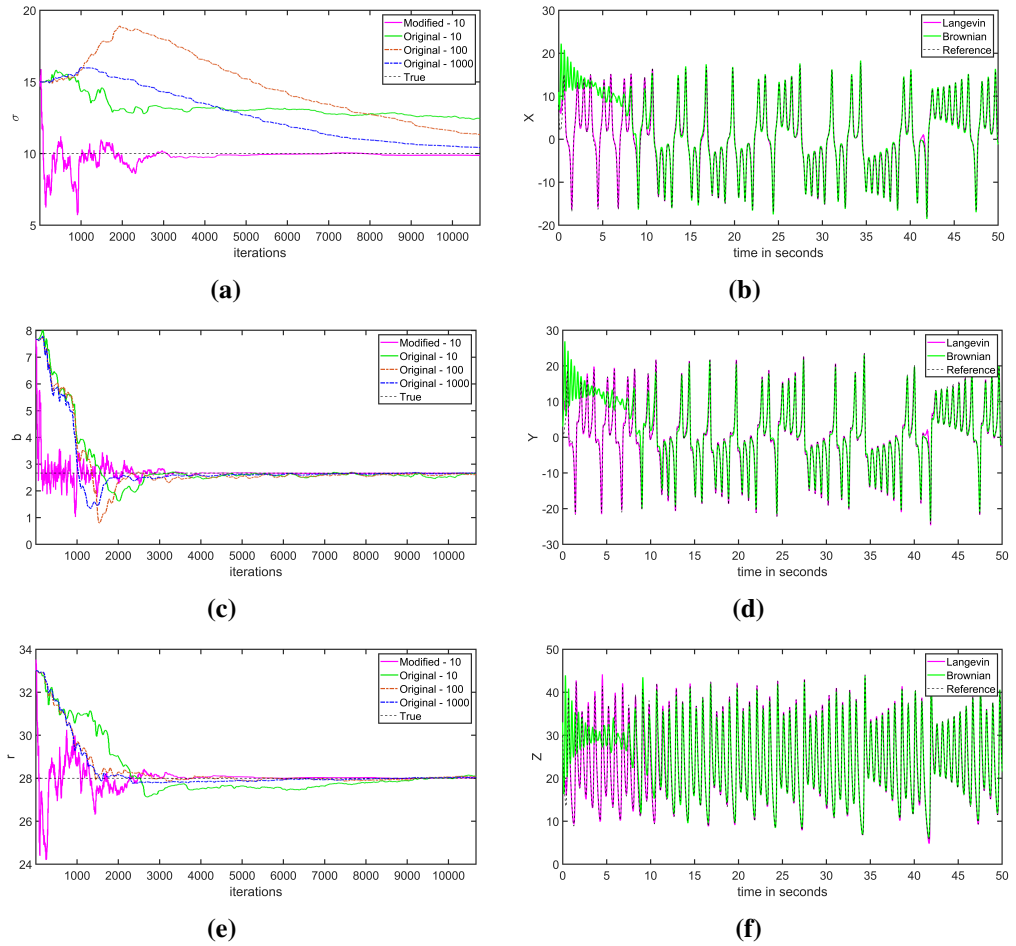


Figure 5.3: Lorenz 63 model: Comparison of filtered states and parameters between the two parameter dynamics, i.e. Brownian (original) and Langevin (modified). Figures (a), (c) and (e) show the convergence of parameters. The numbers accompanying the labels represent the ensemble size used in the EnKF. Figures (b), (d) and (f) show the convergence of the three states for an ensemble size of 10. Data is assimilated only every tenth time step.

the modified parameter dynamics than with the conventional approach. As can be seen from Figure 5.3, subfigures (a), (c) and (e), although the performance of both approaches deteriorates, the Brownian motion suffers more than the proposed dynamics. In particular, the parameter σ fails to converge even with an ensemble size of 1000 (dashed blue line). It may also be observed that the quality of convergence of the states in subfigures (b), (d) and (e) is also affected due to that of the parameters. This happens because - as also discussed earlier in the chapter - the states and parameters evolve in a correlated manner. It is worth pointing out that the gradient term in the Langevin dynamics at any time step is a function of the observations,

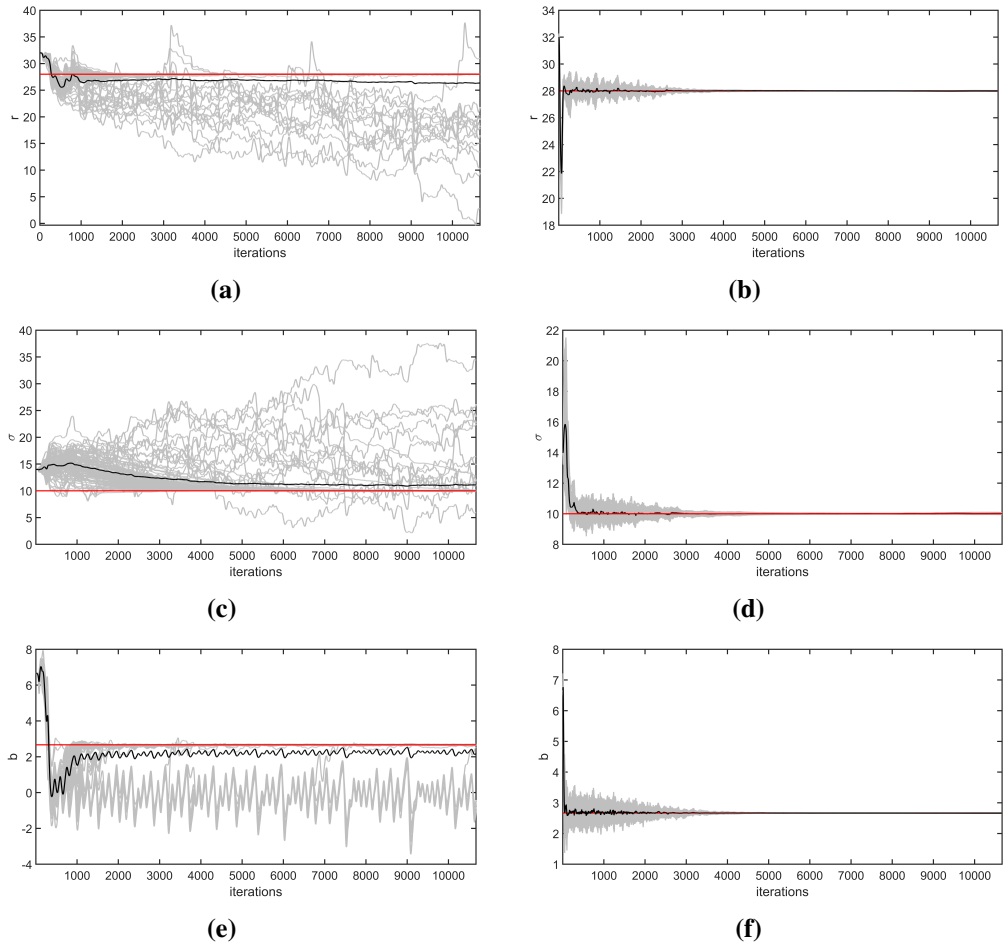


Figure 5.4: Lorenz 63 model: This figure shows sampling variance of filtered parameters for the two parameter dynamics. Figures (a), (c) and (e) are with the Brownian dynamics, while figures (b), (d) and (f) are with the Langevin dynamics for parameters. Each of the grey lines are the results from one independent runs of the EnKF starting with the same initial condition; there are 100 such independent runs. For each of the 100 experiments, the ensemble size used is 25, and data is assimilated at every time step. The black lines represent the means of the independent runs. The red lines represent the true parameter values.

and since the observations are available only every tenth time step, the gradient term is kep t unchanged until a new observation becomes available. Therefore, the fluctuation in the parameter values is only due to the noise term in the Langevin dynamics at time steps when measurements are not available. In real world problems, it is often not possible to control how much data is available. Through this set of experiments, we may observe that the proposed approach makes better use of the data that is available for a given problem. It may also be observed, that simply in-

creasing the ensemble size with Brownian motion for the parameter dynamics does little to improve the results.

The final set of experiments is performed with a view to study the robustness of the proposed method. This is done by repeating 100 times, the same experiment as for Figure 5.2, for an ensemble size of 25. Figure 5.4 shows the results for only the parameters. The subfigures (a), (c) and (e) are the results with Brownian motion while those in (b), (d) and (f) for the Langevin dynamics. The initial point is kept the same to preclude its effect on the results. As is clear from the figures, the proposed method is more robust than the Brownian motion. Also, there is a small bias in the means of the results (solid black lines) with Brownian motion, which is absent for the proposed method. Based on all the three experiments, it may not be unreasonable to envisage this comparison becoming sharper when the frequency of data is lower.

5.5.2 Elasticity model

We apply the parameter dynamics on the identification of the Young's modulus parameter for a linear elastic material model:

$$\sigma = \varepsilon Y \tag{5.12}$$

Hence, the material model mimics purely elastic material behaviour and therefore comes with only a single parameter Y (the Young's modulus) to be identified via filtering procedure. The Young's modulus Y as stiffness measure links strains ε (deformation measure) to stresses σ (force measure) via the relation 5.12. Basically, this linear elastic model behaves like a spring system whereas the spring constant c is set to fit the behaviour seen in experiments under the presumption $F = cu$, as F being the response force of a spring with the spring constant c which is stretched about u. Returning from this analogy, we seek for the stiffness properties of a solid material by identifying its Young's modulus Y from tensile test experiments.

While the identification of the Young's Modulus is generally a simple task, we presently wish to demonstrate the capabilities of filtering for this task. It can

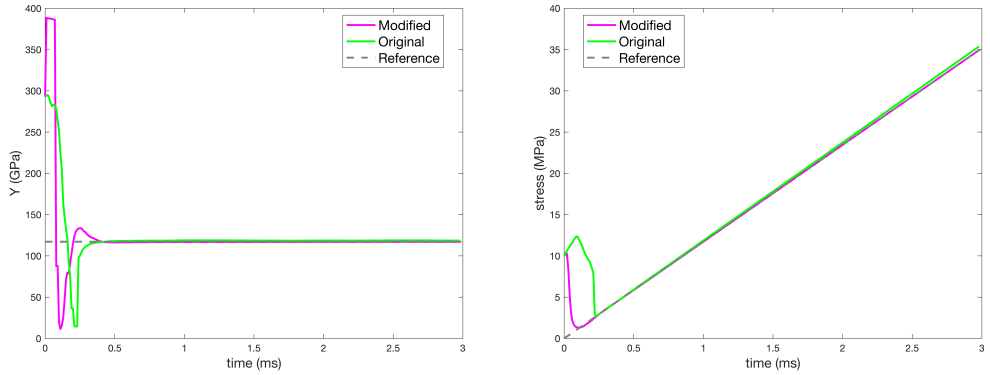
be further applied to the identification of parameters for plastic hardening or the evolution of damage in more sophisticated models, which is left for future work.

We also show that this method is robust for parameter identifications against uncertainties in the starting point by varying the initial stress level. In other words, test stand based strain shifts of a test data set do not need to be corrected (unlike in conventional methods) and the filtering algorithm is still able to recover the correct parameter. This may facilitate test post-processing and test data preparation for a real world application. For demonstration, we use artificial test data, generated with a material model following the formulation 5.12. The filtering procedure takes given values of strain ε and returns the state variable σ (being the only variable within the vector X_i) based on noisy parameter suggestions for the Young's modulus Y (then part of the parameter vector θ_i).

Figure 5.5 shows the parameter convergence from an arbitrary offset of the starting point. Here, the stress level is set to $\sigma = 10MPa$ as initial condition, as depicted in (b). The quantity being measured is the stress σ. The problem parameters used are as follows: ensemble size =5, process noise intensity $\sigma_x = 0.001$, measurement noise intensity $\sigma_z = 0.005$, parameter noise intensity for Brownian motion case $\sigma_\theta = 10^4$, whereas for the Langevin case, the initial annealing parameter $\beta_0 = 5$ which is decayed with time steps as per the following relation $\beta_{k+1} = \beta_k / \exp(0.01k)$. Figure 5.5(a) shows how the algorithm converges the Young's modulus Y from an overestimation due to the initial variable offset, towards the reference value. Figure 5.6 shows results for the same problem as in Figure 5.5, except now the data is assimilated only every tenth time step. In both diagrams it can be understood how the original Brownian motion dynamics are outperformed by PEDAL, the modified Langevin dynamics within filtering.

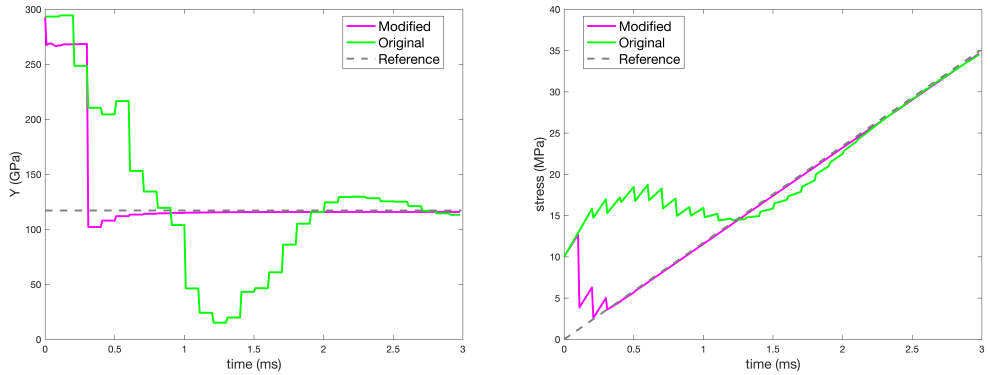
5.5.3 Tsunami propagation

DA for tsunami has been investigated in [148] wherein the tsunami heights and velocity are estimated by assimilating information only from the tsunami wave height measurements obtained in real-time and not the seismic source or tsunami source parameters. In this chapter, we demonstrate that the assimilation of tsunami wave



(a) **(b)**

Figure 5.5: Elasticity model: Comparison of filtered states and parameters between the two parameter dynamics, i.e. Brownian (original) and Langevin (modified). Figure (a) show the convergence of the Young's modulus of elasticity (Y). Figure (b) shows the convergence of the stress (σ). Data is assimilated at every time step.



(a) **(b)**

Figure 5.6: Elasticity model: Comparison of filtered states and parameters between the two parameter dynamics, i.e. Brownian (original) and Langevin (modified). Figure (a) show the convergence of the Young's modulus of elasticity (Y). Figure (b) shows the convergence of the stress (σ). Data is assimilated only every tenth time step.

height (*i.e.* free surface elevation) is greatly aided by joint-estimation of tsunami source parameters. Here, the tsunami is generated using an initial swell in sea surface. After time $t=0$, the raised water falls under the influence of gravity and propagates in the domain. The swell is defined by 5 parameters *viz.*, the amplitude of the swell (A), the location co-ordinates of the centre of the swell (X_o, Y_o), and the length of its semi-axes (L, W). A general tsunami source is a combination of multiple swells/sub-sources, indexed by the superscript ^{i}, *e.g.* A^i amplitude of swell

i. The initial condition to the tsunami propagation is the linear combination of the swells from all the sub-sources.

Sensors are placed in the direction of the wave and sample the wave at regular intervals (*i.e.* assimilation interval δt_a). The data assimilation is initiated when the tsunami wave height at any of the sensors reaches a detection threshold. The number of sensors n_s, and their locations $\left(x_s^j, y_s^j\right)$, where j varies from 1 to n_s, are critical to discriminating between the multiple swells. A judicious positioning can help in reducing the non-uniqueness of the estimated parameters, by curtailing regions of the parameter space[149, 150]. We use 4 sensors placed at the corners of a square, sufficiently separated from each other. At every assimilation instant $t = t_k^a$, the states and parameters are jointly updated using the sensor data at that instant. The model is run from $t = 0$ (unlike the previous examples, wherein the model is run only from the previous time step, i.e. t_k) with new initial conditions corresponding to the updated parameters till the next assimilation instant at $t = t_{k+1}^a$. The process is repeated for each assimilation step.

The 2-D linear long-wave (LLW) model governing tsunami evolution in deep ocean (that we use for our numerical toy experiments) is defined by the following equations [148]:

$$\frac{\partial \eta(x,y,t)}{\partial t} = -\frac{\partial M(x,y,t)}{\partial t} - \frac{\partial N(x,y,t)}{\partial t} \tag{5.13}$$

$$\frac{\partial M(x,y,t)}{\partial t} = -gD(x,y)\frac{\partial \eta(x,y,t)}{\partial t} \tag{5.14}$$

$$\frac{\partial N(x,y,t)}{\partial t} = -gD(x,y)\frac{\partial \eta(x,y,t)}{\partial t} \tag{5.15}$$

$$\tag{5.16}$$

where $\eta(x,y,t)$ is the tsunami height at the location (x,y) at time t, M and N are the two vertically integrated horizontal velocity components in the x and y directions respectively, g is the constant of acceleration due to gravity and $D(x,y)$ is the spatial field of sea-depth. The expression for the initial swell, i.e. $\eta(x,y,0)$ in terms of the

5 parameters as described earlier is:

$$\eta(x,y) = A\left(1+\frac{1}{2}\cos\left(\frac{\pi(x-X_0)}{L}\right)\right)\left(1+\frac{1}{2}\cos\left(\frac{\pi(y-Y_0)}{W}\right)\right) \tag{5.17}$$

We perform two numerical experiments, results for which are shown in Figures 5.7 through 5.12. For the first experiment, we consider 5 subsources, with true parameter values as shown in Table 5.1. See Figure 5.7 for a graphical representation of this source (red blobs), along with the location of the 4 sensors (blue circles). All parameters are kept known except for the swell amplitudes (A), i.e. there are 5 unknown parameters to be solved for via filtering, results for this are shown in Figure 5.8 (The negative sign in some of the curves may be ignored, as only the magnitudes of the swell amplitudes are of relevance). The different colours represent different parameters and different marker size represent the two methods (large: Brownian, small: Langevin). The number label at the end of each line plot refers to the parameter number, i.e. A for subsources 1 through 5. For the second experiment, we take only one source (see Figure 5.10), but assume the amplitude (A), length (L) and width (W) as the unknown parameters. We thus solve for 3 unknown parameters in this case, see Figure 5.11. The number labels here are 1 for A, and 2 and 3 for L and W respectively. Figures 5.9 and 5.12 show a comparison between the measurements at the sensors and the assimilated tsunami heights via the two approaches for the first and second examples, respectively. The problem parameters used for both these experiments are as follows: ensemble size =10, process noise intensity $\sigma_x = 0.001$, measurement noise intensity $\sigma_z = 0.003$, parameter noise intensity for Brownian motion case $\sigma_\theta = 0.1$, whereas for the Langevin case, the initial annealing parameter $\beta_0 = 0.01$ which is decayed with time steps as per the following relation $\beta_{k+1} = \beta_k / \exp(0.001k)$.

This particular tsunami problem is still a work in progress, and perhaps a few comments are in order. Even though there are 5 unknown parameters in the first example, it is an easier one compared to the second, since the effect of the unknown amplitude on the dynamics is linear (see equation 5.17). The little difference in the performance of the two methods is most probably due to the following reasons.

First, since the assimilation of parameter means starting from the beginning with a better initial condition, the problem becomes easier to solve with every iteration when compared with the two other examples. Second, since the system model is linear, EnKF is capable of performing well even with Brownian motion, thus not leaving much scope for improvement. Third, unlike the previous two examples, the ratio of unknown number of parameters and states is quite different, it being 1 to 5 parameters (for each subsource) and 400 states for the tsunami problem vs 3 parameters and 3 states for the Lorenz oscillator and, 1 parameter and 1 state for the elasticity example. The proposed approach only affects the dynamics of the parameters and since that proportion is very small for the tsunami example, the effect of the proposed parameter dynamics seems negligible. As part of the future work, we intend to modify the problem so that we invert for the bathymetry parameters (which can be modelled by the same equation i.e. 5.17) which will now represent a fixed system parameters instead of 'pseudo-parameters' representing an initial state. For this modified problem, it will also be possible to bypass the issue of starting the prediction step from the very beginning, and hence expect a better comparison between the Brownian and proposed Langevin dynamics for parameter evolution.

Subsource #	**A**	X_0	Y_0	**L**	**W**
Parameters for the first example					
1	5	35	15	10	10
2	10	30	20	10	10
3	5	25	25	10	10
4	10	20	30	10	10
5	5	15	35	10	10
Parameters for the second example					
1	5	25	25	10	10

Table 5.1: True values of the tsunami source parameters for both the examples considered.

5.6 Concluding remarks

In this chapter, a novel approach for the evolution of parameters is proposed within the framework of combined state-parameter estimation via DA. An energy-like term

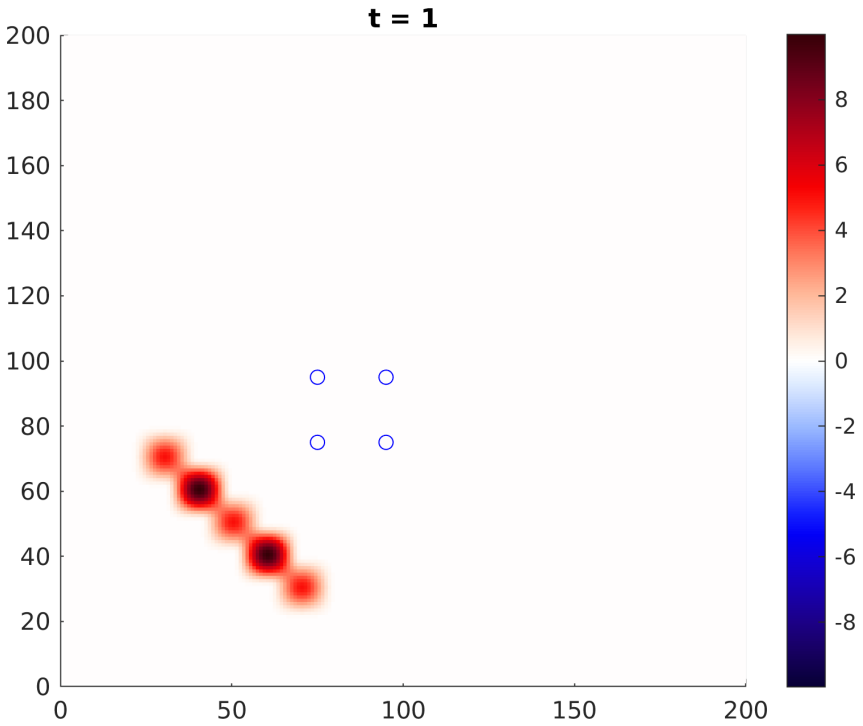


Figure 5.7: Tsunami source representation for the first example. The coloured blobs are a 2-dimensional representation of the 3-dimensional initial tsunami surface. The blue circles represent the locations of the 4 observation sensors.

is proposed as a function of the innovation and the intuition behind this choice discussed. It is worth mentioning again that the form of energy used in this work might have been chosen differently. Any convex function of the innovation might be a reasonable choice. Hence, there is flexibility to choose a form of energy that is more suitable for a given problem. It may be observed that the more chaotic a system is, the more is the advantage of using this approach. This is not unreasonable because in a chaotic system, a small change in the initial states may lead to relatively larger changes later on and hence if Brownian motion is used for parameter dynamics, it slows the convergence of parameters and implicitly also that of the states since both states and parameters co-evolve.

The method is demonstrated with the help of three numerical examples. It may be concluded that when compared with the conventional method of evolving the parameters as per Brownian motion, the results with the proposed approach exhibit considerably faster convergence, better accuracy and reduced sampling variance. It is often difficult to have control over how much data is available. The examples

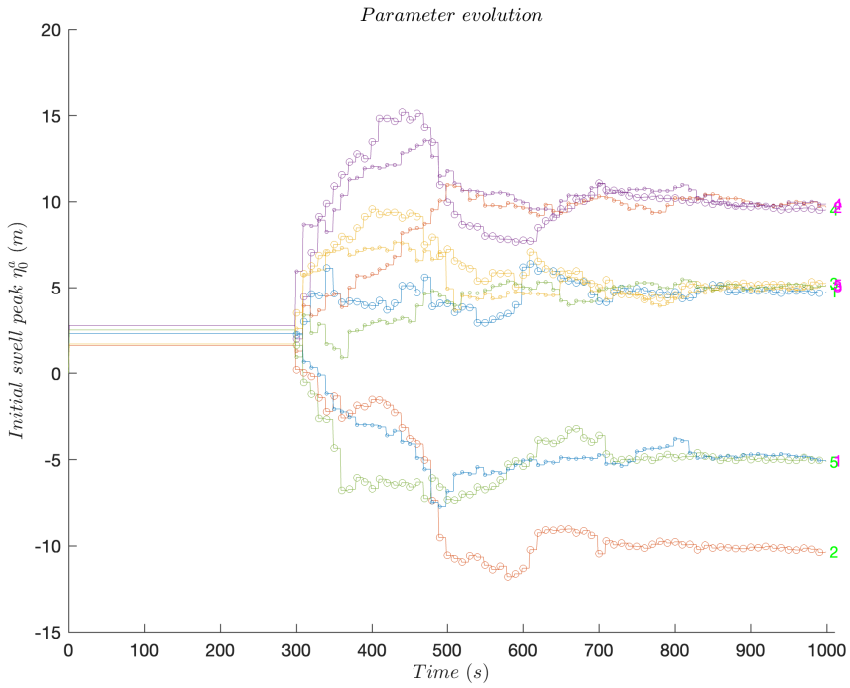


Figure 5.8: Comparison of filtered parameters (ignoring the negative signs) between the two parameter dynamics for the first example. The different colours represent different parameters. Paths with a smaller marker size are the results via PEDAL, whereas ones with a larger marker size are those via Brownian motion. The labels accompanying each path represent the subsource number as per Table 5.1.

also show that when the frequency of data is lower, the contrast between the performances of the conventional and proposed approaches is more pronounced. Thus, it may be concluded that the proposed approach is capable of better exploiting the available data. Overall, in all the examples considered, it can be observed that, with the proposed method, the convergence of states also improves as a consequence of improved parameter convergence. It is worth mentioning at this point, that although there is some increase (about 17 % for the Lorenz 63 model) in the computational cost owing to the calculation of the gradient term in the Langevin dynamics, it is more than compensated with the improvement in the results. More specifically, for the Lorenz oscillator problem, even with an ensemble size of more than 100 times, the conventional approach performs poorly as compared to the proposed approach. Moreover, the proposed approach is parallelisable over the ensemble.

The obesrvations above are all based on rigorous experiments with numerical examples. It would be interesting to study the theoretical convergence properties of the combined state-parameter estimation via filtering after modifying the dynamics as proposed in this chapter. This, however, is a major undertaking and outside the remit of the present work. The [151, 152, 153] offer an analysis of the convergence properties of the EnKF for a large ensemble limit. This could perhaps be a good starting point for undertaking a similar study with the proposed approach. One may also consider other possible variations of this approach. One such possibility is to use constrained Brownian motion for problems where there are physical constraints on the parameters, which is not uncommon. For example, one may use the generalized Langevin dynamics [154], which is not a Markov process, but takes into account the history of the process which helps with numerical stability of solutions. It is also possible to use mirrored Langevin dynamics [155] or stochastically developed Langevin dynamics as in chapter 1 of this thesis with the objective function replaced by the energy in equation (5.7).

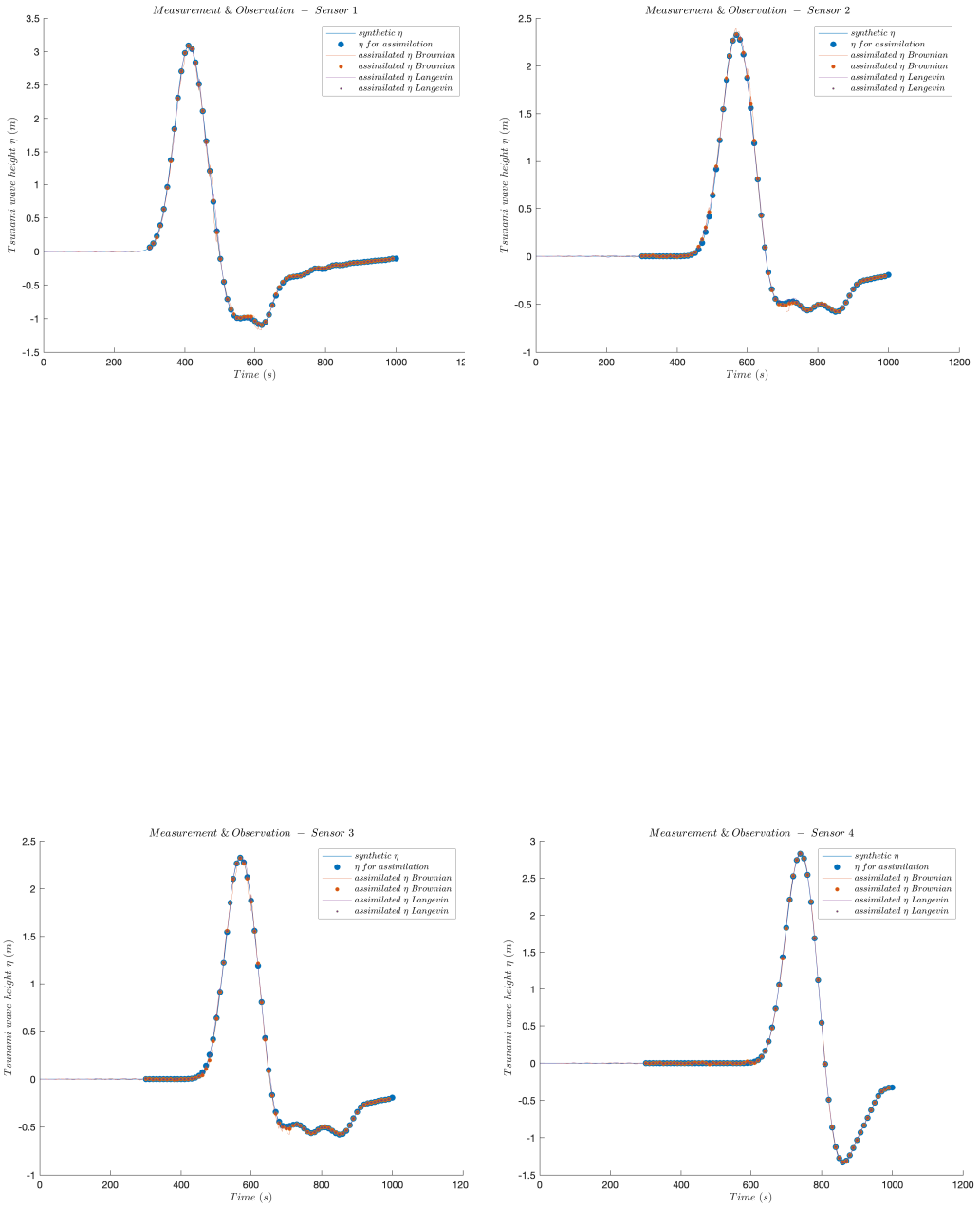


Figure 5.9: Each subplot represents the tsunami height evolution at each of the 4 sensors for the first example. Within each subplot, the three paths represent the following: (blue) observations of the tsunami heights at the sensor, (orange) filtered tsunami height via Brownian dynamics for parameters and (purple) filtered tsunami heights via Langevin dynamics.

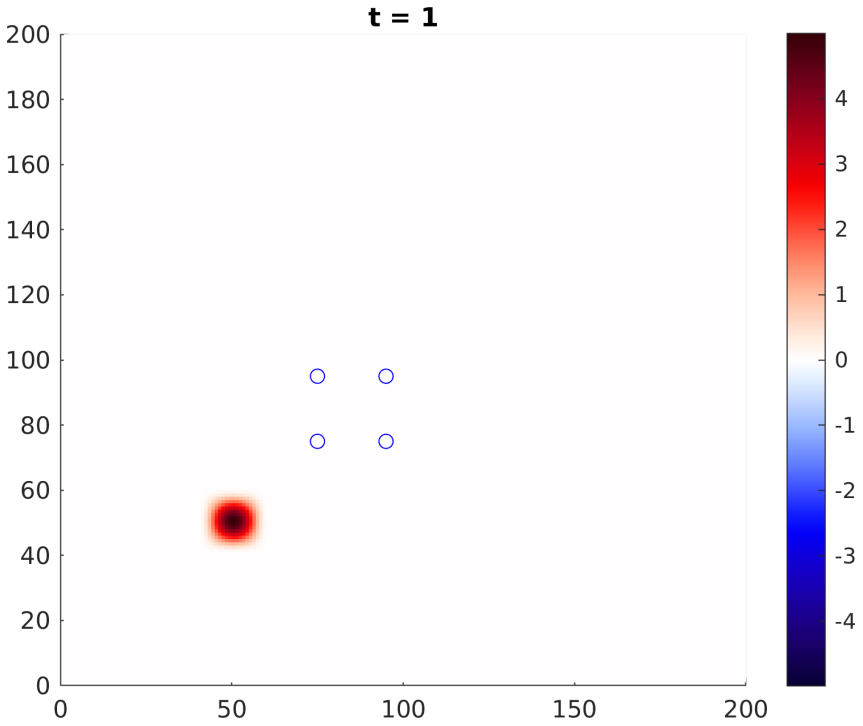


Figure 5.10: Tsunami source representation for the second example. The coloured blob is a 2-dimensional representation of the 3-dimensional initial tsunami surface. The blue circles represent the locations of the 4 observation sensors.

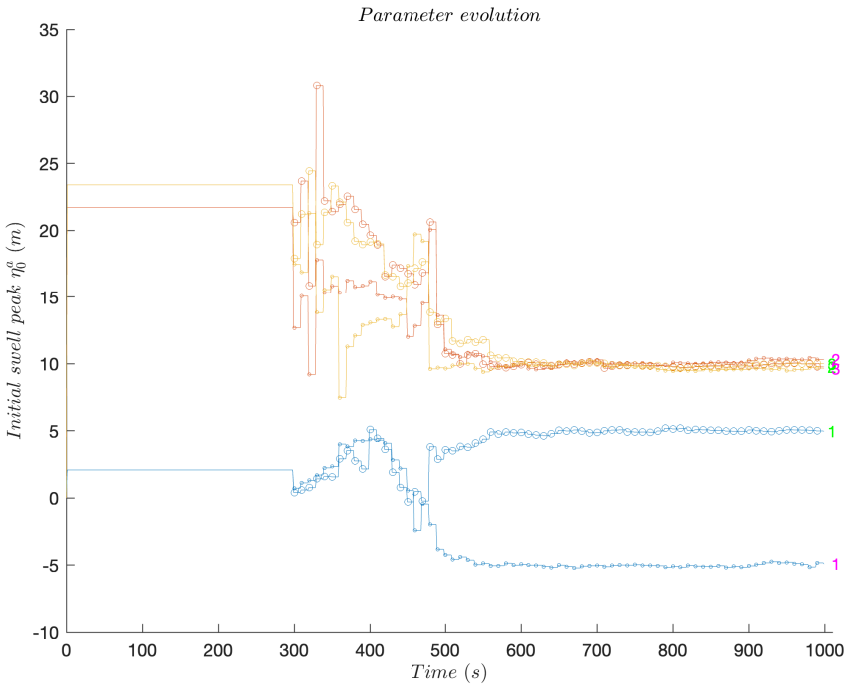


Figure 5.11: Comparison of filtered parameters between the two parameter dynamics for the second example. The different colours represent different parameters. Paths with a smaller marker size are the results via PEDAL, whereas ones with a larger marker size are those via Brownian motion. The labels 1 through 3 represent the A (negative signs to be ignored), L and W parameters respectively.

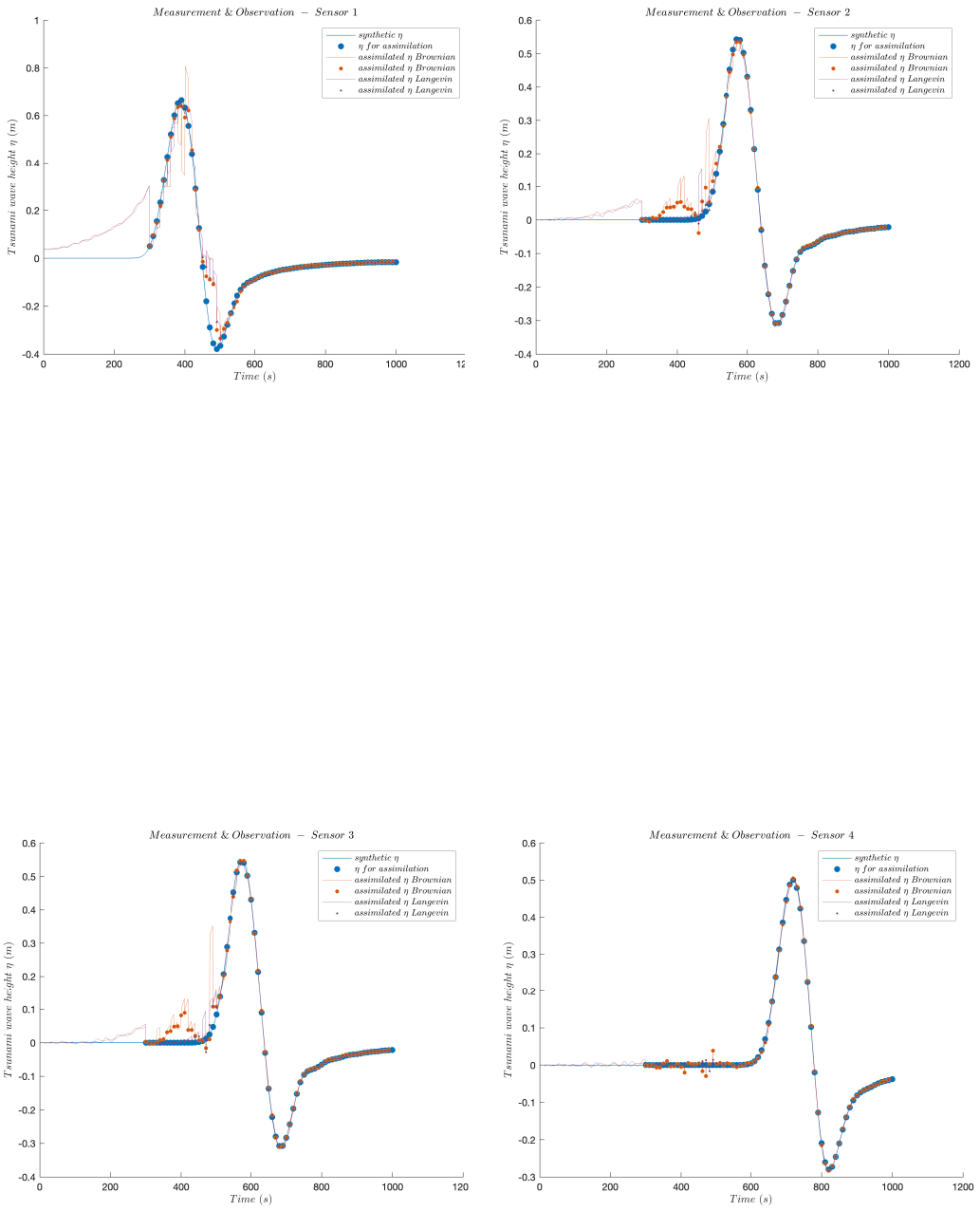


Figure 5.12: Each subplot represents the tsunami height evolution at each of the 4 sensors for the second example. Within each subplot, the three paths represent the following: (blue) observations of the tsunami heights at the sensor, (orange) filtered tsunami height via Brownian dynamics for parameters and (purple) filtered tsunami heights via Langevin dynamics.

Chapter 6

Conclusions and future directions

Novel algorithms for non-convex optimization, Markov chain Monte Carlo (MCMC), stochastic optimal control (SOC) and parameter estimation via data assimilation are proposed in this thesis. These are largely based on concepts in stochastics and Riemannian differential geometry. In chapters 2 through 5, each of these algorithms are derived, discussed and their performances compared with other competing methods with the help of a few numerical examples. In what follows, a few remarks reflecting on these methods are made alongside a few possible future directions of research.

Stochastically developed Langevin dynamics: derivation and application to non-convex optimization

In Chapter 2, a global non-convex optimization method for an unconstrained objective function is proposed. It is based on the stochastically devleoped overdamped Langevin diffusion equation (with annealing) which is the main result of this chapter. This result is derived via three different approaches. The first is based on the concept of stochastic development as in Section 2.2.4. The second is for a special case where a $(d-1)$-dimensional surface is embedded in a d-dimensional space. As detailed in Section 2.3, this is done for a deterministic equation and shown equivalent to the deterministic counterpart of the first approach as per equation 2.12. Finally, in Appendix A, it is proved using Cartan's structure equations and largely follows [156]. With this approach, geometric arguments leading to a modified drift

in the dynamics given by equation A.1 are based purely on deterministic considerations. This chapter also proposes a possible choice of metric for a given objective function.

This is perhaps the first time a seamless approach to global optimization is proposed to deal with the exploration and exploitation trade-off using both stochastics and differential geometry simultaneously. It is probably the first time also that a Riemannian manifold (RM) approach is proposed for an unconstrained and non-convex optimization problem. To the best of our knowledge, RM methods of optimization are so far constructed as counterparts of deterministic Newton and quasi-Newton schemes for constrained convex optimization problems, where the constraint surface itself defines the RM .

The performance of the proposed method is compared with the overdamped Langevin diffusion equation 2.47 as well as the equation existing in the literature for the Langevin diffusion on RM (equation 2.50). As can be observed from the figures in Section 2.4, although the results via equation 2.47 converge to the correct solutions for 2-dimensional Ackley and Rastrigin functions, the method fails for the 40-dimensional cases for both these functions. Even for the 2-dimensional problems, the quality of convergence is poorer compared to that of the results obtained by the proposed method. On the other hand, the results via equation 2.50 fail for both the examples considered for 2-dimensional as well as the 40-dimensional cases.

One disadvantage of the proposed method is that derivatives are required unlike most evolutionary optimization methods that are derivative free. Although only unconstrained non-convex optimization problems are considered in this work, it is possible to extend this approach to constrained problems with the help of, for instance, Lagrange multipliers. It is also possible to work with the generalized Langevin diffusion equation on an RM, which unlike the overdamped Langevin diffusion consists of an acceleration term, see [154] for a reference to this equation. Another possible direction for future research is the design of a better metric than the one used as the choice of the metric is crucial to the performance of the proposed

scheme.

Stochastically developed Langevin dynamics: application to Markov chain Monte Carlo

Chapter 3 showcases the application of equation 2.27 in Chapter 2 to MCMC, a method we refer to as GALA, wherein the Riemannian metric is now the Fisher-Information matrix (FIM). A pseduo-code for GALA is provided in Section 3.2, followed by the significance of using FIM as a metric for problems in statistical parameter estimation. More specifically, given i.i.d. samples from a certain distribution, the class of problems considered consists of estimating parameters of that distribution. The equation we use in this work is at odds with a reported equation for MCMC on RM in the literature, viz. equation 2.50.

We discuss a possible flaw in all existing manifold MCMC methods in section 3.4. We also discuss scenarios when proposals for a certain method for a given problem reduce to proposals for another method for the same problem. Despite the apparent similarity in the proposal equations of various second order methods for MCMC, there is a basic difference in the underlying dynamics that is commonly overlooked. More specifically, the source of second derivative appearing in the proposal equation may be due to one of the following reasons - the Ozaki discretization scheme, the space dependent volatility of the Langevin diffusion or the change in the notion of distances in the case of dynamics on a Riemannian manifold. This last one is the source of the second order derivative in our proposal step of the MCMC.

Through various basic probability distributions, we demonstrate the performance of GALA with other competing methods. It can be seen from the results in Section 4.4 GALA outperforms the other methods considered. We also discuss the issues arising while using Stan for MCMC simulations in Section 3.6. It is perhaps worth pointing out that Stan converges to incorrect values for most parameters in the multivariate Gaussian and the logistic regression problems. Moreover, the comparison between GALA and other methods becomes more pronounced as the dimension of the problem increases, where often GALA is the only successful method. The value of dt we have used for GALA is often one or two magnitudes

higher than that used for MALA. The cost of computation for GALA (which is a second-order method) also scales better with dimension as compared to NUTS (which is a first-order method), see Figure 3.6.

Time recursive control of stochastic dynamical systems using forward dynamics and applications

Chapter 4 is about SOC for a linear-quadratic problem (see Section 4.2) wherein we derive a PDE with an initial condition whose solution gives the solution of the control problem. More importantly, we provide a way to efficiently solve this partial differential equation (PDE) via two coupled SDEs. This is unlike the Hamilton-Jacobi-Bellman (HJB) equation (for solving an SOC problem) which is a PDE with a terminal cost and its solution in terms of SDEs requires solving forward-backward SDEs (FBSDEs). In the process, while the total cost to be minimized and the control objectives remain unchanged, the pathwise solutions could be different from those in the standard FBSDE approaches. This is because the terminal cost in our approach is, to put it loosely, smeared over the entire time interval of interest in order to do away with the terminal condition for the HJB equation and instead obtain an appropriate initial condition.

After a thorough review of stochastic optimal control including the approach based on FBSDEs (see Algorithm 6), we provide a detailed derivation of the solution of optimal control via two different approaches, viz. via the PDEs in cost-to-reach and cost-to-go in Sections 4.3.1 and 4.3.3 respectively. A pseudo-code for the proposed method is also presented for convenience, see Algorithm 3. Also, as we have a specific interest in the case of chaotic oscillators wherein the equation of motion is not a straightforward SDE, the derivation for this class of systems is performed explicitly in Section 4.3.2.

The results with the proposed method are compared with those of an FBSDE method in Figure 4.1 only for a simple 1-dimensional problem. For all other examples considered, the cost of computation for the FBSDE method becomes prohibitive. The most important example is perhaps the Duffing-Holmes oscillator, as the control objective therein is to remain at an unstable fixed point. More specifi-

cally, due to the chaotic nature of the system, it is very easy to shoot far away from the fixed point very quickly even when the initial point is very close to the fixed point, see the uncontrolled trajectory in Figure 4.7. Besides a significant reduction in the computational cost (from $\mathcal{O}(n^2+n)$ to $\mathcal{O}(n)$), the present approach has other advantages. First is, it can be applied to problems where the terminal time is not known. Second, it is possible to start the control at an intermediate time after the system has started evolving. This is particularly beneficial for chaotic systems as discussed in Chapter 4 and may help significantly reduce the high initial cost as seen in the figures in Section 4.4.

As mentioned earlier, a possible future direction is to employ the stochastically developed SDE derived in Chapter 2. The Riemannian metric may be defined by treating the total cost functional as a pseudo-energy as was done for optimization in Chapter 2 and then taking its derivatives with respect to the instantaneous control u_t. Although not straightforward in terms of implementation, this should lead to a method of SOC using just one SDE instead of two.

Efficient parameter estimation via data assimilation: a modified parameter dynamics

Finally, in Chapter 5 we proposed a stochastic data assimilation scheme with a novel parameter dynamics (which we refer to as PEDAL) that may be used instead of the Brownian motion. For clarity, we first discuss state estimation as well as combined state and parameter estimation via ensemble Kalman filter (EnKF) in Sections 5.2 and 5.3 respectively. In Section 5.3, the Brownian motion is used for parameter dynamics, which we replace by a stochastic differential equation (SDE) with a non-trivial drift in Section 5.4. More specifically, we propose a Langevin diffusion equation based on a pseudo-energy function. We also suggest a possible choice for this pseudo-energy in equation 5.7, which is by no means unique. Any convex function of the innovation vector would perhaps be a reasonable choice.

We demonstrate the performance of this approach with three examples. The first is Lorenz 63 oscillator, which is a chaotic system with 3 states and 3 parameters. We observed that PEDAL is faster and more accurate for the states as well

as parameters. It may be observed from Figure 5.2, that even with an ensemble size two orders of magnitude higher, the performance of EnKF is not on a par with PEDAL. On the other hand, when the frequency of assimilation is lower, EnKF may not even converge for all states and parameters, see Figure 5.3. We also demonstrate the robustness of PEDAL by repeated experiments in Figure 5.4. The second example is of a simple linear elasticity model that relates stress and strain. This is a 1-state 1-parameter system. Note that even a linear process model may become a non-linear problem of filtering when the parameters are also unknown. Observations for this case are similar to the previous example. The final example is estimating source parameters of a tsunami. Again, we compare the performance of EnKF with PEDAL. It may be observed from the equations of tsunami evolution, that the first tsunami example is similar in nature to the linear elasticity model, and hence it would not be unreasonable to expect a similar comparison between PEDAL and EnKF. However, this example does not naturally fall in the class of problems for combined state and parameter estimation via data assimilation. The reason being, the parameters we are interested in only define the initial tsunami surface, i.e. the initial state vector, and are not parameters that appear in the tsunami evolution. We discuss the results and possible reasons there is little difference between the performance of the two approaches for this particular example towards the end of Section 5.5.3. It may be noted that the contrast in performance between the PEDAL and EnKF is perhaps the most stark in the case of chaotic systems and/or systems where the dynamics is sensitive to changes in parameter values.

It is worth re-emphasizing, that although the EnKF is used in the present work, it is possible to use any filtering approach since the proposal for parameter dynamics only affects the posing of the filtering problem. This may still be considered a work-in-progress as there are various ways a pseudo-energy may be constructed and also several ways in which the dynamics itself maybe chosen. For instance, the stochastically developed Langevin diffusion equation as derived in Chapter 2 could be a natural choice for better performance. The metric may be defined as the Hessian of the pseudo-energy function chosen. Indeed, this has already been tested for

the Lorenz model. However, since the Langevin diffusion itself converges very fast for this example, there is little scope left for improvement and hence little noticeable difference in the performance with the Hessian as the metric. The tsunami problem presently uses a linear model which in our future work, we intend to replace with the non-linear shallow water model, preferably perhaps with a physics-informed neural network [157] trained to solve the non-linear model to avoid the cumulative time integration error.

Although a mere possibility at this stage, it should be possible to use the stochastically developed SDE for the design of a data assimilation method itself. After all, it is also a problem in sampling from the posterior. The challenge is that the posterior probability in a data assimilation problem evolves with time while in a conventional sampling problem, the posterior probability from which samples are desired remains frozen. The evolution of probability however has a structure. It depends on the dynamics of the system and of the measurement. Indeed, the Kusher-Stratonovich equation (see Chapter 6 in [26]) represents precisely that. There are difficulties in solving this equation due to the presence of a certain circularity. However, if it were possible to solve it iteratively so that the error in distributions, for instance the Kullback-Leibler divergence, between the estimated evolution of probability and the true one was minimized over time, it would perhaps lead to a new perspective with a data assimilation problem. With the help of physics-informed neural networks, stochastic neural networks and the power of differential geometry in stochastics, it seems possible, at least in principle, to develop a data assimilation method for the most general case, i.e. non-linear process and measurement dynamics as well as the presence of non-Gausssian noises in both, which remains to this day an open problem.

Possible future directions

I would like to conclude this thesis by stating a few, perhaps a little far-fetched at this time, directions for future research. Although this thesis has used concepts from Riemannian differential geometry, stochastics and physics, it is merely a start.

Beginning perhaps with the work of Einstein, it is well-known that the physical theories have an umbilical connection with differential geometry. The emergence of gauge theory and its momentous applications in physics are perhaps a case in point. It thus seems worthwhile to use concepts from advanced physics and exploit new symmetries to derive novel forms of Lagrangian and Hamiltonian dynamics towards a more advanced suite of algorithms for problems in optimization or statistical inference. For instance, one may include the Riemannian scalar curvature as an additional term in the Hamiltonian to arrive at a new family of dynamical systems to be integrated for Hamiltonian Monte Carlo simulation. Langevin dynamics may similarly be enhanced with the Ricci curvature flow. Such modifications, generally aimed at certain computational benefits in the search process, introduce certain geometric incompatibilities (homological structures) in the time-continuous setting. Integrating the underlying differential equations in time using the machinery of discrete exterior calculus and thus naturally satisfying these homologies in the time-discrete setting also appears to be an interesting study for the future.

Appendix A

An Appendix for Chapter 2

A.1 Stochastic development with Cartan's structure equations

In what follows, we discuss a method for intrinsically developing a stochastic differential equation from a d-dimensional Euclidean space to a Riemannian manifold M of the same dimension. This approach exploits the notion of an orthonormal frame bundle $F(M)$ on M. Here, every point x in M is furnished with an orthonormal frame Q that serves as an isomorphism between the Euclidean space $\mathbb{R}^d$ and T_xM, the d-dimensional tangent space to M at the point x. The procedure that we adopt largely follows the article by [156] and may be considered both an alternative and extension of the procedure explored in [8] to reach the same result. To start with consider the following SDE in $\mathbb{R}^d$

$$dW_t = \alpha(W_t)dt + dB_t \tag{A.1}$$

where W_t is an $\mathbb{R}^d$-valued stochastic process with α and β being the drift and diffusion fields respectively. B_t is an $\mathbb{R}^d$-valued Brownian motion with independently evolving scalar components. Before we proceed further, let us also recall from [23] the standard equation for the Brownian motion B_t on a Riemannian manifold M in

terms of the standard Euclidean Brownian motion $\bar{B}_t$ which is given by

$$dB_t^i = \left(\sqrt{G^{-1}}\right)_{ij} d\bar{B}_t^j - \frac{1}{2} G_{jk}^{-1} \gamma_{jk}^i dt \tag{A.2}$$

We now need a representation for the vector field α developed on M and this is what we do next, based on the original work by [158] extended later by [159] and [156]. This approach makes use of the Cartan's structure equations to arrive at an appropriate representation of an Euclidean vector field on M. Towards this, let us define a smooth path $x := x_{t_0:t}$ on M and let α be a vector field in $\mathbb{R}^d$. The directional derivative at the point $x(t)$ along α defined as $D_\alpha(x(t))$ is given by $D_\alpha(x(t)) = Q(x(t))\alpha$. By $y_\alpha^x(s)$ we define the integral curve (flow) of the vector field $D_\alpha(x(t)))$, i.e. $\left.\frac{\partial y_\alpha^{x(t)}(s)}{\partial s}\right|_{s=0} = D_\alpha(x(t))$, where $y_\alpha^{x(t)}(0) = x(t)$. Since we are interested in parallel representations of curves in $\mathbb{R}^d$ and M, we also use the symbol $\mathscr{I}$ to relate the two representations. In other words, $\bar{x} = \mathscr{I}^{-1}x$ is the representation in $\mathbb{R}^d$ of the curve x in M. Accordingly, we have

$$\bar{y}_\alpha^{\bar{x}(t)}(s) = \mathscr{I}^{-1} \circ y_\alpha^{x(t)}(s) \circ \mathscr{I} \tag{A.3}$$

This defines a corresponding pushforward map

$$\left.\frac{\partial \bar{y}_\alpha^{\bar{x}(t)}(s)}{\partial s}\right|_{s=0} = (\mathscr{I}_*^{-1} D_\alpha)(\bar{x}(t)) = \mathscr{E}_\alpha(\bar{x}(t)) \tag{A.4}$$

The RHS of the last equation is clearly a vector field in $\mathbb{R}^d$ along $\bar{x}$ which we refer to as $\mathscr{E}_\alpha(\bar{x})$ and for which we wish to arrive at a representation. Towards this, define a canonical 1-form ϕ in $T^*O(M)$ such that for any vector field Z in $T(O(M))$, we have $\phi(Z) = Q^{-1}\pi_*(Z)$, where π_* is the push-forward of the canonical projection $\pi : O(M) \to M$. To proceed further, we now make use of Cartan's structure equations given as follows.

$$d\phi = -\omega \wedge \phi + \Theta \tag{A.5}$$

$$d\omega = -\omega \wedge \omega + \Omega \tag{A.6}$$

In the equations above, $\wedge$ is the skew wedge product of differential forms and ω denotes the $o(d)$-valued connection 1-forms, i.e. a $d \times d$ skew-symmetric matrix with each element being a 1-form. Θ is an $\mathbb{R}^d$-valued torsion 2-form which is identically zero for a Riemannian manifold. Ω is the $o(d)$-valued curvature 2-form. Since we are dealing with both the curves $x(t)$ which is parametrized in t and $y_\alpha^{x(t)}(s)$ parametrized in s and starting at $x(t)$, we may consider a frame Q to be a function of both t and s, i.e. $Q \equiv Q(t,s)$. This enables us to write the following velocities

$$T = \frac{\partial Q(t,s)}{\partial t}, \quad S = \frac{\partial Q(t,s)}{\partial s}, \quad N = \frac{\partial \bar{y}_\alpha^{\bar{x}(t)}(s)}{\partial t} \tag{A.7}$$

which yields the following identification

$$\mathscr{E}_\alpha(\bar{x}(t)) = \int_0^t \left(\frac{\partial N(\tau,s)}{\partial s}\bigg|_{s=0} \right) d\tau \tag{A.8}$$

Clearly, $Q(t,s)$ for a fixed t is the horizontal lift of $\bar{y}_\alpha^{\bar{x}(t)}(s)$ on $O(M)$, so that we have $T = HN$, where H is a horizontal vector field. This is also equivalent to $N = \phi(T)$ which leads to the following upon differentiation with respect to s.

$$\frac{\partial N}{\partial s} = S\phi(T) \tag{A.9}$$

At this stage we invoke the following formula for exterior differentiation. For two vector fields T and S , and the closed 2-form $d\phi$, we have

$$d\phi(T,S) = T\phi(S) - S\phi(T) - \phi([T,S]) \tag{A.10}$$

where, the Lie bracket $[T,S]$ is presently zero since the time like co-ordinates s and t are chosen independently. Hence, we have

$$S\phi(T) = T\phi(S) - d\phi(T,S) \tag{A.11}$$

Moreover, by observing that $\pi(Q(t,0)) = y_\alpha^{x(t)}(0) = x(t)$, we directly have $\pi_*(S) = Q(t,0)\alpha$. In other words, from this we retrieve the horizontal component of S as

$H\alpha$, which is equivalent to the following

$$\phi(S) = \alpha \tag{A.12}$$

This yields

$$\left.\frac{\partial N(t,s)}{\partial s}\right|_{s=0} = \dot{\alpha} - d\phi(T,S) \tag{A.13}$$

where overdot denotes derivative with respect to 't'. We need to simplify $d\phi(T,S)$ in the equation above using the two structure equations of Cartan. Using the first one, we immediately have

$$d\phi(T,S) = \omega(S)N \tag{A.14}$$

Note that, S is not necessarily a purely horizontal vector field unlike T, i.e. we have $\omega(T) = 0$ since the connection ω is a purely vertical 1-form. Now, using the second structure equation (A.6) and the formula for exterior differentiation [160], we get

$$d\omega(T,S) = T\omega(S) = \Omega(T,S) \tag{A.15}$$

Integrating the last equation over t, we arrive at the required expression for the connection 1-form $\omega(S)$

$$\omega_{Q(t,0)}(S) = \int_0^t \Omega_{Q(\tau,0)}(T,S)d\tau \tag{A.16}$$

Note that the curvature 2-form Ω is strictly horizontal. This, along with the fact that $T = H\left.\frac{\partial \bar{y}_\alpha^{\bar{x}(t)}(s)}{\partial t}\right|_{s=0} := H\dot{\bar{y}}_\alpha^{\bar{x}(t)}(0)$ leads to

$$\omega_{Q(t,0)}(S) = \int_0^t \Omega_{Q(\tau,0)}(H\dot{\bar{y}}_\alpha^{\bar{x}(t)}(0), H\alpha)d\tau := \int_0^t \mathscr{K}_\alpha(\tau)d\tau \tag{A.17}$$

Substituting (A.17) in (A.14) and putting this back in (A.13), we have

$$\left.\frac{\partial N(t,s)}{\partial s}\right|_{s=0} = \dot{\alpha} - \left(\int_0^t \mathscr{K}_\alpha(\tau)d\tau\right)\dot{\bar{y}}_\alpha^{\bar{x}(t)}(0) \tag{A.18}$$

See the Appendix A.1.1 for the expression for $\mathscr{K}_\alpha$. Integrating once more with

respect to t, we get

$$\mathscr{E}(\bar{x}(t)) = \alpha - \int_0^t \int_0^t \mathscr{K}_\alpha(\tau) d\tau d\bar{y}_\alpha^{\bar{x}(\tau)}(0) = \alpha - \int_0^t \int_0^t \mathscr{K}_\alpha(\tau) d\tau d\bar{x}(\tau) \quad \text{(A.19)}$$

$\mathscr{K}_\alpha$ is clearly a matrix with scalar entries which are functions of τ. Therefore, restricting the double integral in (A.19) to $[t, t+\Delta t]$, we observe that the integral is of the order $(\Delta t)^{\frac{3}{2}}$, provided $\bar{x}(t)$ is a Brownian motion in $\mathbb{R}^d$. Hence, from the perspective of numerical integration, it constitutes a higher order term which is ignored in this work. With this approximation in place, we may transfer the developed vector field $\mathscr{E}(\bar{x}(t))$ from $\mathbb{R}^d$ to T_xM to get $Q_{(t,0)}\alpha$, which is the modified drift in the stochastically developed SDE that we shall make use of in this work. This additional drift, when added to the equation for Brownian motion on an RM (A.2), should lead to the equation for a general SDE on the RM. At this stage, we need a representation of Q in terms of the Riemannian metric tensor g, which is given by $Q = \sqrt{g^{-1}}$, see chapter 3 of [23]. The developed SDE corresponding to (A.1) thus takes the form

$$dx_t^i = \left[\sqrt{g^{-1}(x_t)}\right]_{ij} \alpha^j(x_t)dt - \frac{1}{2}\left[g^{-1}(x_t)\right]_{kl} \gamma_{kl}^i(x_t)dt + \left[\sqrt{g^{-1}(x_t)}\right]_{im} dB_t^m \quad \text{(A.20)}$$

A.1.1 Expression for $\mathscr{K}_\alpha$

H in equation (A.17) is the horizontal vector field given by

$$H_i = Q_i^j X_j - Q_i^j Q_m^l \gamma_{jl}^k X_{km} \quad i \in [1,d] \quad \text{(A.21)}$$

where $X_i = \frac{\partial}{\partial x^i}$ and $X_{km} = \frac{\partial}{\partial Q_m^k}$ are the basis of the tangent space of the frame bundle $F(M)$. Therefore, each vector in equation (A.17) of the form Hv where v lies in $\mathbb{R}^d$ may be written as

$$Hv = v^i Q_i^j X_j - v^i Q_i^j Q_m^l \gamma_{jl}^k X_{km} \quad \text{(A.22)}$$

For convenience, let $\dot{\bar{y}}_{\alpha}^{\bar{x}(t)}(0) = \eta$, then the integrand in equation (A.17) may be simplified in terms of the curvature $(3,1)$-tensor R^{i}_{jkl} as

$$
\begin{aligned}
(\mathscr{K}_{\alpha})^{c}_{d} &= \frac{1}{2} R^{c}_{dab} dx^{a} \wedge dx^{b}(H\eta, H\alpha) \qquad \text{(A.23)} \\
&= \frac{1}{2} R^{c}_{dab} dx^{a} \wedge dx^{b}(\eta^{i} Q^{j}_{i} X_{j} - \eta^{i} Q^{j}_{i} Q^{l}_{m} \gamma^{k}_{jl} X_{km}\,, \alpha^{p} Q^{q}_{p} X_{q} - \alpha^{p} Q^{q}_{p} Q^{s}_{t} \gamma^{k}_{qs}(x) X_{rt}) \\
&= \frac{1}{2} R^{c}_{dab} (\eta^{i} Q^{j}_{i} \delta^{a}_{j} \alpha^{p} Q^{q}_{p} \delta^{b}_{q} - \eta^{i} Q^{j}_{i} \delta^{b}_{j} \alpha^{p} Q^{q}_{p} \delta^{a}_{q}) \\
&= \frac{1}{2} R^{c}_{dab} (\eta^{i} Q^{a}_{i} \alpha^{p} Q^{b}_{p} - \eta^{i} Q^{b}_{i} \alpha^{p} Q^{a}_{p}) \\
&= \frac{1}{2} R^{c}_{dab} (\eta^{i} Q^{a}_{i} \alpha^{p} Q^{b}_{p} - \eta^{i} Q^{b}_{i} \alpha^{p} Q^{a}_{p})
\end{aligned}
$$

A.2 Derivatives of Ackley function

$$
f(\mathbf{x}) = f(x_1, ..., x_d) = -a \exp\left(-b \sqrt{\frac{1}{d} \sum_{i=1}^{d} x_i^2} \right) - \exp\left(\frac{1}{d} \sum_{i=1}^{d} \cos(c x_i) \right) + a + \exp(1)
$$

Treat $f(x)$ as an energy-like function to determine g and γ. The Langevin SDE to be developed is given by:

$$
dX_t = -\beta_t \nabla f(X_t) dt + \sqrt{2\beta_t} dB_t
$$

where β is an annealing like parameter. The developed SDE is:

$$
dX_t = -\sqrt{g^{-1}} \beta_t \nabla f(X_t) dt + \sqrt{g^{-1}} \beta_t dB_t - \frac{1}{2} g^{-1} \gamma dt
$$

where

$$
g_{ij} = \frac{1}{2} \frac{\partial^2 f(x)}{\partial x_i \partial x_j} + \Upsilon \delta_{ij} \qquad \text{(A.24)}
$$

Let

$$
T_1(x) = \exp\left(-b \sqrt{\frac{1}{d} \sum_{i=1}^{d} x_i^2} \right)
$$

$$
T_2(x) = \exp\left(\frac{1}{d} \sum_{i=1}^{d} \cos(c x_i) \right)
$$

Therefore, g and its derivatives can be written as

$$g_{ij} = -\frac{a}{2}\frac{\partial^2 T_1(x)}{\partial x_j \partial x_k} - \frac{1}{2}\frac{\partial^2 T_2(x)}{\partial x_j \partial x_k}$$

and

$$\frac{\partial g_{ij}}{\partial x_m} = -\frac{a}{2}\frac{\partial^3 T_1(x)}{\partial x_j \partial x_k \partial x_k} - \frac{1}{2}\frac{\partial^3 T_2(x)}{\partial x_j \partial x_k \partial x_k}$$

We need the first, second and third order derivatives of T_1, T_2.

First derivative of T_1

$$\begin{aligned}\frac{\partial T_1(x)}{\partial x_j} &= \frac{\partial \exp(-\frac{b}{\sqrt{d}}(x_i x_i)^{\frac{1}{2}})}{\partial x_j} \\ &= -\frac{b}{\sqrt{d}}(x_i x_i)^{-\frac{1}{2}} x_j T_1(x)\end{aligned}$$

Second derivative of T_1

$$\begin{aligned}\frac{\partial^2 T_1(x)}{\partial x_j \partial x_k} &= \frac{\partial}{\partial x_k}\left(-\frac{b}{\sqrt{d}}(x_i x_i)^{-\frac{1}{2}} x_j T_1(x)\right) \\ &= \frac{b}{\sqrt{d}} x_j x_k (x_i x_i)^{-\frac{3}{2}} T_1(x) - \frac{b}{\sqrt{d}}(x_i x_i)^{-\frac{1}{2}} \delta_{jk} T_1(x) - \frac{b}{\sqrt{d}}(x_i x_i)^{-\frac{1}{2}} x_j \frac{\partial (T_1(x))}{\partial x_k}\end{aligned}$$

Third derivative of T_1

$$\begin{aligned}
\frac{\partial^3 T_1(x)}{\partial x_j \partial x_k \partial x_m} &= \frac{b}{\sqrt{d}}\frac{\partial}{\partial x_m}\left(x_j x_k (x_i x_i)^{-\frac{3}{2}} T_1(x)\right) - \frac{b}{\sqrt{d}}\frac{\partial}{\partial x_m}\left((x_i x_i)^{-\frac{1}{2}}\delta_{jk} T_1(x)\right) \\
&\quad -\frac{b}{\sqrt{d}}\frac{\partial}{\partial x_m}\left((x_i x_i)^{-\frac{1}{2}} x_j \frac{\partial (T_1(x))}{\partial x_k}\right) \\
&= \frac{b}{\sqrt{d}} x_k (x_i x_i)^{-\frac{3}{2}} T_1(x)\delta_{jm} + \frac{b}{\sqrt{d}} x_j (x_i x_i)^{-\frac{3}{2}} T_1(x)\delta_{km} \\
&\quad -3\frac{b}{\sqrt{d}} x_j x_k x_m (x_i x_i)^{-\frac{5}{2}} T_1(x) + \frac{b}{\sqrt{d}} x_j x_k (x_i x_i)^{-\frac{3}{2}} \frac{\partial T_1(x)}{\partial x_m} \\
&\quad \frac{b}{\sqrt{d}}\delta_{jk} x_m (x_i x_i)^{-\frac{3}{2}} T_1(x) - \frac{b}{\sqrt{d}}(x_i x_i)^{-\frac{1}{2}}\delta_{jk}\frac{\partial T_1(x)}{\partial x_m} \\
&\quad +\frac{b}{\sqrt{d}} x_j x_m (x_i x_i)^{-\frac{3}{2}} \frac{\partial T_1(x)}{\partial x_k} - \frac{b}{\sqrt{d}}(x_i x_i)^{-\frac{1}{2}}\frac{\partial T_1(x)}{\partial x_k}\delta_{jm} - \frac{b}{\sqrt{d}} x_j (x_i x_i)^{-\frac{1}{2}}\frac{\partial^2 T_1(x)}{\partial x_k \partial x_m}
\end{aligned}$$

First derivative of T_2

$$\begin{aligned}
\frac{\partial T_2(x)}{\partial x_j} &= \frac{\partial}{\partial x_j}\left(\exp\left(\frac{1}{d}\sum_{i=1}^{d}\cos(cx_i)\right)\right) && \text{(A.25)} \\
&= -\frac{c}{d}\sin(cx_j) T_2(x) && \text{(A.26)}
\end{aligned}$$

Second derivative of T_2

$$\begin{aligned}
\frac{\partial^2 T_2(x)}{\partial x_j \partial x_k} &= -\frac{c}{d}\frac{\partial(\sin(cx_j) T_2(x))}{\partial x_k} \\
&= -\frac{c^2}{d}\delta_{jk}\cos(cx_j) T_2(x) - \frac{c}{d}\sin(cx_j)\frac{\partial T_2(x)}{\partial x_k}
\end{aligned}$$

Third derivative of T_2

$$\begin{aligned}
\frac{\partial^3 T_2(x)}{\partial x_m \partial x_j \partial x_k} &= \frac{\partial}{\partial x_m}\left(-\frac{c^2}{d}\delta_{jk}\cos(cx_j) T_2(x) - \frac{c}{d}\sin(cx_j)\frac{\partial T_2(x)}{\partial x_k}\right) \\
&= \frac{c^3}{d}\delta_{km} T_2(x)\sin(cx_j) - \frac{c}{d}\sin(cx_j)\frac{\partial^2 T_2(x)}{\partial x_k \partial x_m} \\
&\quad -\frac{c^2}{d}\delta_{jk}\cos(cx_j)\frac{\partial T_2(x)}{\partial x_m} - \frac{c^2}{d}\delta_{jm}\frac{\partial T_2(x)}{\partial x_k}(\cos(cx_j)) \quad \text{(A.27)}
\end{aligned}$$

A.3 Derivatives of Rastrigin function

$$f(\mathbf{x}) = f(x_1,...,x_d) = \sum_{i=1}^{d} [b(x_i)^2 - a\cos(2\pi x_i)] + a \times d \tag{A.28}$$

$$\frac{\partial f}{\partial x_k} = 2bx_k + 2\pi a \sin(2\pi x_k) \tag{A.29}$$

$$\frac{\partial^2 f}{\partial x_k x_n} = \delta_{kn}[2b + 4\pi^2 a\cos(2\pi x_k)] \tag{A.30}$$

$$\frac{\partial^3 f}{\partial x_k x_n x_r} = -a \times 8\pi^3 \delta_{kn}\delta_{nr}\sin(2\pi x_k)[15pt] \tag{A.31}$$

Thus g and its derivatives may be written as

$$g_{ij} = \frac{1}{2}\frac{\partial^2 f(x)}{\partial x_i \partial x_j} + \Upsilon\delta_{ij} \tag{A.32}$$

$$\frac{\partial g_{ij}}{\partial x_k} = \frac{1}{2}\frac{\partial^3 f(x)}{\partial x_i \partial x_j \partial x_k} \tag{A.33}$$

Appendix B

An Appendix for Chapter 3

B.1 Invariant distribution of simplified MMALA

The proposal SDE for simplified MMALA is

$$dX_t = \frac{1}{2}G^{-1}(X_t)\nabla \log(\pi(X_t))dt + \sqrt{G^{-1}(X_t)}dB_t \tag{B.1}$$

The Fokker-Planck equation [26] for this SDE is

$$\frac{\partial p(x,t)}{\partial t} = -\frac{\partial}{\partial x}\left(\frac{1}{2}G^{-1}(x)\nabla \log(\pi(x))p(x,t)\right) + \frac{\partial^2}{\partial x^2}\left(\frac{G^{-1}(x)}{2}p(x,t)\right) \tag{B.2}$$

Let $\pi(x)$ be its stationary solution so that the LHS of B.2 vanishes. Accordingly, substituting $p(x,t) = \pi(x)$ on the RHS, we have

$$\begin{aligned}
& -\frac{\partial}{\partial x}\left(\frac{1}{2}G^{-1}(x)\nabla \log(\pi(x))\pi(x)\right) + \frac{\partial^2}{\partial x^2}\left(\frac{G^{-1}(x)}{2}\pi(x)\right) \\
= \quad & -\frac{1}{2}\frac{\partial}{\partial x}\left(G^{-1}(x)\frac{\partial \pi(x)}{\partial x}\right) + \frac{\pi(x)}{2}\frac{\partial^2 G^{-1}(x)}{\partial x^2} + \frac{G^{-1}(x)}{2}\frac{\partial^2 \pi(x)}{\partial x^2} \\
= \quad & -\frac{1}{2}\frac{\partial G^{-1}(x)}{\partial x}\frac{\partial \pi(x)}{\partial x} - \frac{1}{2}G^{-1}(x)\frac{\partial^2 \pi(x)}{\partial x^2} + \frac{\pi(x)}{2}\frac{\partial^2 G^{-1}(x)}{\partial x^2} + \frac{G^{-1}(x)}{2}\frac{\partial^2 \pi(x)}{\partial x^2} \\
= \quad & -\frac{1}{2}\frac{\partial G^{-1}(x)}{\partial x}\frac{\partial \pi(x)}{\partial x} + \frac{\pi(x)}{2}\frac{\partial^2 G^{-1}(x)}{\partial x^2}
\end{aligned} \tag{B.3}$$

The RHS does not vanish, meaning the assumption of $\pi(x)$ being the invariant distribution of the simplified MMALA SDE is incorrect.

B.2 Invariant distribution of MMALA

For the SDE in equation B.4, the Fokker-Planck equation on Riemannian manifold is given by equation B.5 [73]

$$dx^i = a^i dt + \sigma^{ij} dB_j \tag{B.4}$$

$$\frac{\partial p}{\partial t} = \frac{1}{2\sqrt{g}} \frac{\partial}{\partial x^i} \left(\sqrt{g} g^{ij} \frac{\partial p}{\partial x^j} \right) - \frac{1}{\sqrt{g}} \frac{\partial}{\partial x^i} \left(p\sqrt{g} \left[a^i + \frac{1}{2} g^{jk} \gamma^i_{jk} \right] \right) \tag{B.5}$$

The equation above assumes $\sigma^{im}(\sigma^T)^{mj} = g^{ij}$ .The drift term in MMALA is $a^i = \frac{g^{ij}}{p}\frac{\partial p}{\partial x^j} - \frac{1}{2} g^{jk}\Gamma^i_{jk}$, substituting in B.5, we have

$$\begin{aligned} \frac{\partial p}{\partial t} &= \frac{1}{\sqrt{g}} \frac{\partial}{\partial x^i} \left\{ \frac{1}{2}\sqrt{g} g^{ij} \frac{\partial p}{\partial x^j} - p\sqrt{g} \left[\frac{g^{ij}}{p} \frac{\partial p}{\partial x^j} - \frac{1}{2} g^{jk}\Gamma^i_{jk} + \frac{1}{2} g^{jk}\gamma^i_{jk} \right] \right\} \\ &= 0 \end{aligned} \tag{B.6}$$

Thus, the MMALA method converges to the stationary distribution of the Fokker-Planck equation on Riemannian manifolds.

B.3 Rayleigh distribution

Probability density function: $p(x;\sigma) = \frac{x}{\sigma^2} \exp\left(-\frac{x^2}{2\sigma^2}\right)$
Mean: $\sigma\sqrt{\frac{\pi}{2}}$
Variance: $\frac{(4-\pi)\sigma^2}{2}$
Log-likelihood: $L = \log(p(x;\sigma)) = \log(x) - 2\log(\sigma) - \frac{x^2}{2\sigma^2}$
Gradient of log-likelihood: $\frac{\partial L}{\partial \sigma} = -\frac{2}{\sigma} + \frac{x^2}{\sigma^3}$

Fisher-Information matrix:

$$\begin{aligned} G &= E\left(\frac{\partial L}{\partial \sigma}\frac{\partial L}{\partial \sigma}\right) \\ &= E\left(\left(-\frac{2}{\sigma}+\frac{x^2}{\sigma^3}\right)^2\right) \\ &= E\left(\frac{4}{\sigma^2}+\frac{x^4}{\sigma^6}-\frac{4x^2}{\sigma^4}\right) \end{aligned} \tag{B.7}$$

$$\text{where: } E(x^2) = \int_0^\infty (x^2)\left(\frac{x}{\sigma^2}\right)\exp\left(-\frac{x^2}{2\sigma^2}\right) = 2\sigma^2 \tag{B.8}$$

$$E(x^4) = \int_0^\infty (x^4)\left(\frac{x}{\sigma^2}\right)\exp\left(-\frac{x^2}{2\sigma^2}\right) = 8\sigma^4 \tag{B.9}$$

Therefore, $G = E\left(\frac{4}{\sigma^2}+\frac{8\sigma^4}{\sigma^6}-\frac{4\times 2\sigma^2}{\sigma^4}\right) = \frac{4}{\sigma^2}$
Derivative of Fisher-Information matrix: $\frac{\partial G}{\partial \sigma} = -\frac{8}{\sigma^3}$
Connection: $\gamma = G^{-1}\frac{\partial G}{\partial \sigma} = -\frac{\sigma^2}{4}\frac{8}{\sigma^3} = -\frac{2}{\sigma}$

B.4 Banana-shaped distribution

Probability density function: $p(z_1,z_2;B) \propto \exp\left(-\frac{z_1^2}{200}-\frac{1}{2}(z_2+Bz_1^2-100B)^2\right)$
In the above equation z_1 and z_2 are distributed as $\sim \mathcal{N}(0,\Sigma)$, where

$$\Sigma = \begin{bmatrix} 100 & 0 \\ 0 & 1 \end{bmatrix} \tag{B.10}$$

Log-likelihood: $L = \log(p(z_1,z_2;B)) = -\frac{z_1^2}{200}-\frac{1}{2}(z_2+Bz_1^2-100B)^2$
Gradient of log-likelihood: $\frac{\partial L}{\partial B} = -(z_2+Bz_1^2-100B)(z_1^2-100)$ Fisher-Information

matrix:

$$
\begin{aligned}
G &= E\left[\frac{\partial L}{\partial B}\frac{\partial L}{\partial B}\right] \qquad \text{(B.11)}\\
&= E\left[(z_2+Bz_1^2-100B)^2(z_1^2-100)^2\right]\\
&= E\left[(z_2^2+B^2z_1^4+10000B^2+2Bz_1^2z_2-200Bz_2-200B^2z_1^2)(z_1^4+10000-200z_1^2)\right]\\
&= E[(z_1^4z_2^2+B^2z_1^8+10000B^2z_1^4+2Bz_1^6z_2-200Bz_2z_1^4-200B^2z_1^6)\\
&\quad +10000(z_2^2+B^2z_1^4+10000B^2+2Bz_1^2z_2-200Bz_2-200B^2z_1^2)]\\
&\quad -200(z_1^2z_2^2+B^2z_1^6+10000B^2z_1^2+2Bz_1^4z_2-200Bz_1^2z_2-200B^2z_1^4)\\
&= E[(z_1^4z_2^2+B^2z_1^8+10000B^2z_1^4-200B^2z_1^6)+10000(z_2^2+B^2z_1^4+10000B^2-200B^2z_1^2)\\
&\quad -200(z_1^2z_2^2+B^2z_1^6+10000B^2z_1^2-200B^2z_1^4)]\\
&= 3\sigma_1^4\sigma_2^2+7B^2\sigma_1^8+3\times10000B^2\sigma_1^4-200\times5B^2\sigma_1^6\\
&\quad +10000(\sigma_2^2+3B^2\sigma_1^4+10000B^2-200B^2\sigma_1^2)\\
&\quad -200(\sigma_1^2\sigma_2^2+5B^2\sigma_1^6+10000B^2\sigma_1^2-200\times3B^2\sigma_1^4)\\
&= 3\times10^4+7\times10^8B^2+3\times10^8B^2-10^9B^2+10^4+3\times10^8B^2+10^8B^2-2\times10^8B^2\\
&\quad -200(100+5\times10^6B^2+10^6B^2-6\times10^6B^2)\\
&= 3\times10^4+10^4+2\times10^8B^2-2\times10^4\\
&= 2\times10^4+2\times10^8B^2 \qquad \text{(B.12)}
\end{aligned}
$$

G for the product of likelihoods over N observations is $N\times G$ Therefore, Derivative of Fisher-Information matrix: $N\times4\times10^8B$

Connection:

$$
\begin{aligned}
\gamma &= G^{-1}\frac{\partial G}{\partial B}\\
&= [N\times(2\times10^4+2\times10^8B^2)]^{-1}(N\times4\times10^8B)\\
&= \frac{4\times10^8B}{(2\times10^4+2\times10^8B^2)}\\
&= \frac{4\times10^4B}{(2+2\times10^4B^2)}
\end{aligned}
\qquad \text{(B.13)}
$$

B.5 Weibull distribution

Probability density function:

$$\begin{aligned} p(x;\lambda,k) &= \frac{k}{\lambda}\left(\frac{x}{\lambda}\right)^{k-1}\exp^{-\left(\frac{x}{\lambda}\right)^k}; x \geq 0 \\ &= 0 \quad ; \quad x < 0 \end{aligned}$$

Log-likelihood:

$$\begin{aligned} L = \log(p(x;\lambda,k)) &= \log(k) - \log(\lambda) + (k-1)\log(x) - (k-1)\log(\lambda) - \left(\frac{x}{\lambda}\right)^k \\ &= \log(k) - k\log(\lambda) + (k-1)\log(x) - \left(\frac{x}{\lambda}\right)^k \end{aligned}$$

Gradient of log-likelihood:

$$\begin{aligned} \frac{\partial L}{\partial \lambda} &= -\frac{k}{\lambda} - k\left(\frac{x}{\lambda}\right)^{k-1}\frac{(-x)}{\lambda^2} \\ &= -\frac{k}{\lambda} + \frac{k}{\lambda}\left(\frac{x}{\lambda}\right)^k \\ &= \left[\left(\frac{x}{\lambda}\right)^k - 1\right]\frac{k}{\lambda} \\ \frac{\partial L}{\partial k} &= \frac{1}{k} - \log(\lambda) + \log(x) - \left(\frac{x}{\lambda}\right)^k \log\left(\frac{x}{\lambda}\right) \end{aligned} \tag{B.14}$$

Therefore

$$\nabla L = \begin{bmatrix} \left[\left(\frac{x}{\lambda}\right)^k - 1\right]\frac{k}{\lambda} \\ \frac{1}{k} - \log(\lambda) + \log(x) - \left(\frac{x}{\lambda}\right)^k \log\left(\frac{x}{\lambda}\right) \end{bmatrix} \tag{B.15}$$

Fisher-Information matrix:

$$\begin{aligned} G &= E\left(\begin{bmatrix} \frac{\partial L}{\partial \lambda}\frac{\partial L}{\partial \lambda} & \frac{\partial L}{\partial \lambda}\frac{\partial L}{\partial k} \\ \frac{\partial L}{\partial k}\frac{\partial L}{\partial \lambda} & \frac{\partial L}{\partial k}\frac{\partial L}{\partial k} \end{bmatrix}\right) \\ &= \begin{bmatrix} E\left(\frac{\partial L}{\partial \lambda}\frac{\partial L}{\partial \lambda}\right) & E\left(\frac{\partial L}{\partial \lambda}\frac{\partial L}{\partial k}\right) \\ E\left(\frac{\partial L}{\partial k}\frac{\partial L}{\partial \lambda}\right) & E\left(\frac{\partial L}{\partial k}\frac{\partial L}{\partial k}\right) \end{bmatrix} = \begin{bmatrix} G_{11} & G_{12} \\ G_{21} & G_{22} \end{bmatrix} \end{aligned} \tag{B.16}$$

Where

$$G_{11} = E\left(\left[\left(\frac{x}{\lambda}\right)^k - 1\right]^2 (\frac{k}{\lambda})^2\right) \tag{B.17}$$

$$= \left(\frac{k}{\lambda}\right)^2 E\left(\left[\left(\frac{x}{\lambda}\right)^k - 1\right]^2\right)$$

$$G_{12} = E\left(\left[\left(\frac{x}{\lambda}\right)^k - 1\right] \frac{k}{\lambda}\left[\frac{1}{k} - \log(\lambda) + \log(x) - \left(\frac{x}{\lambda}\right)^k \log\left(\frac{x}{\lambda}\right)\right]\right) \tag{B.18}$$

$$= E\left(\left[\left(\frac{x}{\lambda}\right)^k - 1\right] \frac{k}{\lambda}\left(\frac{1}{k} - \log(\lambda)\right)\right)$$

$$+E\left(\left[\left(\frac{x}{\lambda}\right)^k - 1\right] \frac{k}{\lambda}\left(\log(x) - \left(\frac{x}{\lambda}\right)^k \log\left(\frac{x}{\lambda}\right)\right)\right)$$

$$= \frac{k}{\lambda}\left(\frac{1}{k} - \log(\lambda)\right) E\left(\left[\left(\frac{x}{\lambda}\right)^k - 1\right]\right)$$

$$+\frac{k}{\lambda} E\left(\left[\left(\frac{x}{\lambda}\right)^k - 1\right]\left[\log(x) - \left(\frac{x}{\lambda}\right)^k \log\left(\frac{x}{\lambda}\right)\right]\right) = G_{21}$$

$$G_{22} = E\left(\left\{\frac{1}{k} - \log(\lambda) + \log(x) - \left(\frac{x}{\lambda}\right)^k \log\left(\frac{x}{\lambda}\right)\right\}^2\right) \tag{B.19}$$

Derivative of the Fisher-Information matrix

Derivative w.r.t λ:

$$\frac{\partial G}{\partial \lambda} = \begin{bmatrix} \frac{\partial G_{11}}{\partial \lambda} & \frac{\partial G_{12}}{\partial \lambda} \\ \\ \frac{\partial G_{21}}{\partial \lambda} & \frac{\partial G_{22}}{\partial \lambda} \end{bmatrix} \tag{B.20}$$

$$\begin{aligned}
\frac{\partial G_{11}}{\partial \lambda} &= \frac{\partial \left\{ \left(\frac{k}{\lambda}\right)^2 E\left(\left[\left(\frac{x}{\lambda}\right)^k - 1 \right]^2 \right) \right\}}{\partial \lambda} \qquad \text{(B.21)} \\
&= \frac{\partial \left(\frac{k}{\lambda}\right)^2}{\partial \lambda} E\left(\left[\left(\frac{x}{\lambda}\right)^k - 1 \right]^2 \right) + \left(\frac{k}{\lambda}\right)^2 \frac{\partial E\left(\left[\left(\frac{x}{\lambda}\right)^k - 1 \right]^2 \right)}{\partial \lambda} \\
&= \frac{\partial \left(\frac{k}{\lambda}\right)^2}{\partial \lambda} E\left(\left[\left(\frac{x}{\lambda}\right)^k - 1 \right]^2 \right) + \left(\frac{k}{\lambda}\right)^2 E\left(\frac{\partial \left[\left(\frac{x}{\lambda}\right)^k - 1 \right]^2}{\partial \lambda} \right) \\
&= -\frac{2k^2}{\lambda^3} E\left(\left[\left(\frac{x}{\lambda}\right)^k - 1 \right]^2 \right) - \left(\frac{k}{\lambda}\right)^2 E\left(2k \left(\left(\frac{x}{\lambda}\right)^k - 1 \right) \frac{x^k}{\lambda^{k+1}} \right) \\
&= -\frac{2k^2}{\lambda^3} E\left(\left[\left(\frac{x}{\lambda}\right)^k - 1 \right]^2 \right) - \frac{2k^3}{\lambda^{k+3}} E\left[\left(\frac{x}{\lambda}\right)^k - 1 \right] \\
\frac{\partial G_{12}}{\partial \lambda} &= \frac{\partial \left\{ \frac{k}{\lambda}\left(\frac{1}{k} - \log(\lambda)\right) E\left(\left[\left(\frac{x}{\lambda}\right)^k - 1 \right] \right) + \frac{k}{\lambda} E\left(\left[\left(\frac{x}{\lambda}\right)^k - 1 \right] \left[\log(x) - \left(\frac{x}{\lambda}\right)^k \log\left(\frac{x}{\lambda}\right) \right] \right) \right\}}{\partial \lambda} \qquad \text{(B.22)} \\
&= \frac{\partial \left\{ \frac{k}{\lambda}\left(\frac{1}{k} - \log(\lambda)\right) E\left(\left[\left(\frac{x}{\lambda}\right)^k - 1 \right] \right) \right\}}{\partial \lambda} + \frac{\partial \left\{ \frac{k}{\lambda} E\left(\left[\left(\frac{x}{\lambda}\right)^k - 1 \right] \left[\log(x) - \left(\frac{x}{\lambda}\right)^k \log\left(\frac{x}{\lambda}\right) \right] \right) \right\}}{\partial \lambda} \\
&= -\frac{k}{\lambda^2}\left(\frac{1}{k} - \log(\lambda)\right) E\left[\left(\frac{x}{\lambda}\right)^k - 1 \right] - \frac{k}{\lambda}\frac{1}{\lambda} E\left[\left(\frac{x}{\lambda}\right)^k - 1 \right] \\
&\quad + \frac{k}{\lambda}\left(\frac{1}{k} - \log(\lambda)\right) E\left(\frac{\partial \left[\left(\frac{x}{\lambda}\right)^k - 1 \right]}{\partial \lambda} \right) - \frac{k}{\lambda^2} E\left(\left[\left(\frac{x}{\lambda}\right)^k - 1 \right] \left[\log(x) - \left(\frac{x}{\lambda}\right)^k \log\left(\frac{x}{\lambda}\right) \right] \right) \\
&\quad + \frac{k}{\lambda} E\left(\frac{\partial \left[\left(\frac{x}{\lambda}\right)^k - 1 \right]}{\partial \lambda} \left[\log(x) - \left(\frac{x}{\lambda}\right)^k \log\left(\frac{x}{\lambda}\right) \right] \right) \\
&\quad + \frac{k}{\lambda} E\left(\left[\left(\frac{x}{\lambda}\right)^k - 1 \right] \frac{\partial \left[\log(x) - \left(\frac{x}{\lambda}\right)^k \log\left(\frac{x}{\lambda}\right) \right]}{\partial \lambda} \right) \\
&= -\frac{k}{\lambda^2}\left(\frac{1}{k} - \log(\lambda)\right) E\left[\left(\frac{x}{\lambda}\right)^k - 1 \right] - \frac{k}{\lambda}\frac{1}{\lambda} E\left[\left(\frac{x}{\lambda}\right)^k - 1 \right] \\
&\quad + \frac{k}{\lambda}\left(\frac{1}{k} - \log(\lambda)\right) E\left(\frac{\partial \left[\left(\frac{x}{\lambda}\right)^k \right]}{\partial \lambda} \right) - \frac{k}{\lambda^2} E\left(\left[\left(\frac{x}{\lambda}\right)^k - 1 \right] \left[\log(x) - \left(\frac{x}{\lambda}\right)^k \log\left(\frac{x}{\lambda}\right) \right] \right) \\
&\quad + \frac{k}{\lambda} E\left(\frac{\partial \left[\left(\frac{x}{\lambda}\right)^k \right]}{\partial \lambda} \left[\log(x) - \left(\frac{x}{\lambda}\right)^k \log\left(\frac{x}{\lambda}\right) \right] \right) - \frac{k}{\lambda} E\left(\left[\left(\frac{x}{\lambda}\right)^k - 1 \right] \frac{\partial \left[\left(\frac{x}{\lambda}\right)^k \log\left(\frac{x}{\lambda}\right) \right]}{\partial \lambda} \right) \\
&= -\frac{k}{\lambda^2}\left(\frac{1}{k} - \log(\lambda)\right) E\left[\left(\frac{x}{\lambda}\right)^k - 1 \right] - \frac{k}{\lambda}\frac{1}{\lambda} E\left[\left(\frac{x}{\lambda}\right)^k - 1 \right] - \frac{k}{\lambda}\left(\frac{1}{k} - \log(\lambda)\right) E\left(\frac{kx^k}{\lambda^{k+1}} \right) \\
&\quad - \frac{k}{\lambda^2} E\left(\left[\left(\frac{x}{\lambda}\right)^k - 1 \right] \left[\log(x) - \left(\frac{x}{\lambda}\right)^k \log\left(\frac{x}{\lambda}\right) \right] \right) \\
&\quad - \frac{k}{\lambda} E\left(\frac{kx^k}{\lambda^{k+1}} \left[\log(x) - \left(\frac{x}{\lambda}\right)^k \log\left(\frac{x}{\lambda}\right) \right] \right) \\
&\quad - \frac{k}{\lambda} E\left(\left[\left(\frac{x}{\lambda}\right)^k - 1 \right] \left[-\frac{k}{\lambda}\left(\frac{x}{\lambda}\right)^k \log\left(\frac{x}{\lambda}\right) - \left(\frac{x}{\lambda}\right)^k \frac{1}{\lambda} \right] \right)
\end{aligned}$$

$$\begin{aligned}
\frac{\partial G_{22}}{\partial \lambda} &= \frac{\partial [E(\{\frac{1}{k} - \log(\lambda) + \log(x) - (\frac{x}{\lambda})^k \log(\frac{x}{\lambda})\}^2)]}{\partial \lambda} \\
&= E[\frac{\partial (\{\frac{1}{k} - \log(\lambda) + \log(x) - (\frac{x}{\lambda})^k \log(\frac{x}{\lambda})\}^2)}{\partial \lambda}] \\
&= 2E\left[\left\{\frac{1}{k} - \log(\lambda) + \log(x) - \left(\frac{x}{\lambda}\right)^k \log\left(\frac{x}{\lambda}\right)\right\}\left\{\frac{-1}{\lambda} + \left[\frac{k}{\lambda}\left(\frac{x}{\lambda}\right)^k \log\left(\frac{x}{\lambda}\right) + \frac{x^k}{\lambda^{k+1}}\right]\right\}\right]
\end{aligned} \tag{B.23}$$

Derivative w.r.t k:

$$\frac{\partial G}{\partial k} = \begin{bmatrix} \frac{\partial G_{11}}{\partial k} & \frac{\partial G_{12}}{\partial k} \\ \\ \frac{\partial G_{21}}{\partial k} & \frac{\partial G_{22}}{\partial k} \end{bmatrix} \tag{B.24}$$

$$\begin{aligned}
\frac{\partial G_{11}}{\partial k} &= \frac{\partial \{(\frac{k}{\lambda})^2 E([(\frac{x}{\lambda})^k - 1]^2)\}}{\partial k} \\
&= \frac{\partial (\frac{k}{\lambda})^2}{\partial k} E\left[\left(\frac{x}{\lambda}\right)^k - 1\right]^2 + \left(\frac{k}{\lambda}\right)^2 \frac{\partial \left\{E([(\frac{x}{\lambda})^k - 1]^2)\right\}}{\partial k} \\
&= \frac{2k}{\lambda^2} E\left[\left(\frac{x}{\lambda}\right)^k - 1\right]^2 + \left(\frac{k}{\lambda}\right)^2 E \frac{\partial ([(\frac{x}{\lambda})^k - 1]^2)}{\partial k} \\
&= \frac{2k}{\lambda^2} E\left[\left(\frac{x}{\lambda}\right)^k - 1\right]^2 + 2\left(\frac{k}{\lambda}\right)^2 E\left(\left[\left(\frac{x}{\lambda}\right)^k - 1\right] \log\left(\frac{x}{\lambda}\right)\left(\frac{x}{\lambda}\right)^k\right)
\end{aligned} \tag{B.25}$$

$$
\begin{aligned}
\frac{\partial G_{12}}{\partial k} &= \frac{\partial\{\frac{k}{\lambda}(\frac{1}{k}-\log(\lambda))E([(\frac{x}{\lambda})^k-1])\}}{\partial k} \qquad \text{(B.}\\
&\quad +\frac{\partial\{\frac{k}{\lambda}E([(\frac{x}{\lambda})^k-1][\log(x)-(\frac{x}{\lambda})^k\log(\frac{x}{\lambda})])\}}{\partial k}\\
&= \frac{\partial(\frac{1}{\lambda}-\frac{k\log(\lambda)}{\lambda})}{\partial k}E\left[\left(\frac{x}{\lambda}\right)^k-1\right]+\left(\frac{1}{\lambda}-\frac{k\log(\lambda)}{\lambda}\right)\frac{\partial\{E([(\frac{x}{\lambda})^k-1])\}}{\partial k}\\
&\quad +\frac{1}{\lambda}E\left(\left[\left(\frac{x}{\lambda}\right)^k-1\right]\left[\log(x)-\left(\frac{x}{\lambda}\right)^k\log\left(\frac{x}{\lambda}\right)\right]\right)\\
&\quad +\frac{k}{\lambda}E\left(\frac{\partial([(\frac{x}{\lambda})^k-1][\log(x)-(\frac{x}{\lambda})^k\log(\frac{x}{\lambda})])}{\partial k}\right)\\
&= -\frac{\log(\lambda)}{\lambda}E\left[\left(\frac{x}{\lambda}\right)^k-1\right]+\left(\frac{1}{\lambda}-\frac{k\log(\lambda)}{\lambda}\right)E\left[\left(\frac{x}{\lambda}\right)^k\log\left(\frac{x}{\lambda}\right)\right]\\
&\quad +\frac{1}{\lambda}E\left(\left[\left(\frac{x}{\lambda}\right)^k-1\right]\left[\log(x)-\left(\frac{x}{\lambda}\right)^k\log\left(\frac{x}{\lambda}\right)\right]\right)\\
&\quad +\frac{k}{\lambda}E\left(\left(\frac{x}{\lambda}\right)^k\log\left(\frac{x}{\lambda}\right)\left[\log(x)-\left(\frac{x}{\lambda}\right)^k\log\left(\frac{x}{\lambda}\right)\right]-\left[\left(\frac{x}{\lambda}\right)^k-1\right]\left[\left(\frac{x}{\lambda}\right)^k\log(\frac{x}{\lambda})\log\left(\frac{x}{\lambda}\right)\right.\right.\\
&= -\frac{\log(\lambda)}{\lambda}E\left[\left(\frac{x}{\lambda}\right)^k-1\right]+\left(\frac{1}{\lambda}-\frac{k\log(\lambda)}{\lambda}\right)E\left[\left(\frac{x}{\lambda}\right)^k\log\left(\frac{x}{\lambda}\right)\right]\\
&\quad +\frac{1}{\lambda}E\left(\left[\left(\frac{x}{\lambda}\right)^k-1\right]\left[\log(x)-\left(\frac{x}{\lambda}\right)^k\log\left(\frac{x}{\lambda}\right)\right]\right)\\
&\quad +\frac{k}{\lambda}E\left(\left(\frac{x}{\lambda}\right)^k\log\left(\frac{x}{\lambda}\right)\left[\log(x)-2\left(\frac{x}{\lambda}\right)^k\log\left(\frac{x}{\lambda}\right)+\log\left(\frac{x}{\lambda}\right)\right]\right)\\
\frac{\partial G_{22}}{\partial k} &= \frac{\partial[E(\{\frac{1}{k}-\log(\lambda)+\log(x)-(\frac{x}{\lambda})^k\log(k)\}^2)]}{\partial k} \qquad \text{(B.}\\
&= E\frac{\partial(\{\frac{1}{k}-\log(\lambda)+\log(x)-(\frac{x}{\lambda})^k\log(\frac{x}{\lambda})\}^2)}{\partial k}\\
&= 2E\left[\left\{\frac{1}{k}-\log(\lambda)+\log(x)-\left(\frac{x}{\lambda}\right)^k\log\left(\frac{x}{\lambda}\right)\right\}\left\{-\frac{1}{k^2}-\left(\frac{x}{\lambda}\right)^k\log\left(\frac{x}{\lambda}\right)\log\left(\frac{x}{\lambda}\right)\right\}\right]
\end{aligned}
$$

The expectations for this distribution appearing in the expressions for G and derivatives of G are evaluated numerically at every step of the Markov chain (for GALA and MMALA) by sampling from the Weibull distribution with parameters equal to k and λ at each step. Connection: $\gamma^k_{ij} = G^{kl}(\partial_i g_{jl} + \partial_j g_{il} - \partial_l g_{ij})$

B.6 Multivariate Gaussian distribution

Probability density function: $p(y;\mu,\Sigma) = \frac{1}{2\pi}|\Sigma|^{\frac{-1}{2}} \exp(-\frac{1}{2}(y-\mu)^T\Sigma^{-1}(y-\mu))$

Two major derivations are required. Namely, the derivation of **gradient** and the **Hessian** of log-likelihood with respect to the **square root** of covariance matrix.

Log - Likelihood: $\log(p(y;\mu,\Sigma)) = L = -\log(2\pi) - \frac{1}{2}\log(|\Sigma|) - \frac{1}{2}(y-\mu)^T\Sigma^{-1}(y-\mu)$

For a d-dimensional distribution, let

$$\begin{aligned} \theta &= (\theta_1, \theta_2, ..., \theta_D) \text{ , where } D = \frac{d^2+3d}{2} \\ &= (\mu_1, \mu_2, ...\mu_d, \Sigma_{11}, \Sigma_{21}, \Sigma_{22}, \Sigma_{31}, \Sigma_{32}, ..., \Sigma_{d(d-1)}, \Sigma_{dd}) \end{aligned} \tag{B.28}$$

Gradient of Log-likelihood:

Gradient of Log-likelihood with respect to μ

$$\frac{\partial L}{\partial \mu} = \frac{1}{2}[\Sigma^{-1} + (\Sigma^{-1})^T](y-\mu) \tag{B.29}$$

$$\text{i.e. } \frac{\partial L}{\partial \theta_i} = \frac{1}{2}\{[\Sigma^{-1} + (\Sigma^{-1})^T](y-\mu)\}_i \text{ , } i \in [1,d]$$

Gradient of log-likelihood with respect to Σ

$$\frac{\partial L}{\partial \Sigma} = -\frac{1}{2}\frac{\partial(\log(|\Sigma|))}{\partial \Sigma} - \frac{1}{2}\frac{\partial[(y-\mu)^T\Sigma^{-1}(y-\mu)]}{\partial \Sigma} \tag{B.30}$$

$$\text{where} \frac{\partial(\log(|\Sigma|))}{\partial \Sigma} = (\Sigma^{-1})^T \quad [161] \tag{B.31}$$

$$\text{Let } P(\Sigma) = [(y-\mu)^T\Sigma^{-1}(y-\mu)] \tag{B.32}$$

Re-writing in indicial notation as follows

$$\begin{aligned} P(\Sigma) &= (y-\mu)_p\{\Sigma^{-1}(y-\mu)\}_p \\ &= (y-\mu)_p\Sigma^{-1}_{pq}(y-\mu)_q \\ \text{Therefore } \frac{\partial L}{\partial \Sigma} = &= -\frac{1}{2}(\Sigma^{-1})^T - \frac{1}{2}\frac{\partial P(\Sigma)}{\partial \Sigma} \end{aligned} \tag{B.33}$$

$$\begin{aligned}
\text{Now, } \frac{\partial P(\Sigma)}{\partial \Sigma_{ij}} &= \frac{\partial [(y-\mu)_p \Sigma^{-1}_{pq} (y-\mu)_q]}{\partial \Sigma_{ij}} \\
&= (y-\mu)_p \frac{\partial \Sigma^{-1}_{pq}}{\partial \Sigma_{ij}} (y-\mu)_q \\
&= -(y-\mu)_p \Sigma^{-1}_{pi} \Sigma^{-1}_{jq} (y-\mu)_q \\
&= -(\Sigma^{-1})^T_{ip} (y-\mu)_p (y-\mu)_q (\Sigma^{-1})^T_{qj} \\
&= -(\Sigma^{-1})^T_{ip} [(y-\mu)(y-\mu)^T]_{pq} (\Sigma^{-1})^T_{qj} \\
&= -[(\Sigma^{-1})^T [(y-\mu)(y-\mu)^T] (\Sigma^{-1})^T]_{ij}
\end{aligned} \tag{B.34}$$

Thus B.33 can be written in matrix form again as follows

$$\frac{\partial L}{\partial \Sigma} = -\frac{1}{2}(\Sigma^{-1})^T + \frac{1}{2}[(\Sigma^{-1})^T (y-\mu)(y-\mu)^T (\Sigma^{-1})^T] \tag{B.35}$$

Thus, we can evaluate the partial derivatives with respect to θ_{d+1} through θ_D using B.35 together with B.28.

Fisher Information Matrix: $G = E[\,(\partial_i \log p(x;\theta))\,(\partial_j \log p(x;\theta))^T\,]$

From equations B.29 andB.35, we have

$$\frac{\partial \log p(y;\theta)}{\partial \theta} = [\frac{\partial L}{\partial \mu_1}, \frac{\partial L}{\partial \mu_2}, ... \frac{\partial L}{\partial \mu_d}, \frac{\partial L}{\partial \Sigma_{p_1 q_1}}, \frac{\partial L}{\partial \Sigma_{p_2 q_2}} .... \frac{\partial L}{\partial \Sigma_{p_{D-d} q_{D-d}}}]^T \tag{B.36}$$

where $(p_1 q_1)$, $(p_2 q_2)$, $..(p_{D-d} q_{D-d})$ correspond to the pairs of indices of the lower triangular part of the covariance matrix (including diagonals). Thus $\frac{\partial \log p(y;\theta)}{\partial \theta}$ is of the form $[\vec{a_1}\ \vec{a_2}]$ where $\vec{a_1} = \{\frac{\partial L}{\partial \mu}\}_{d\times 1}$ and $\vec{a_2} = vec\{\frac{\partial L}{\partial \Sigma}\}_{(D-d)\times 1}$ Hence,

$$G_{ij} = E[\,(\partial_i \log p(x;\theta))\,(\partial_j \log p(x;\theta))\,] \tag{B.37}$$

Consider the following cases

1. $i, j < d$
2. $i > d,\ j < d$
3. $i < d,\ j > d$
4. $i, j > d$

G_{ij} has to be evaluated separately for each of the above cases.

Case 1. $i, j < d$

$$\begin{aligned} G_{ij} &= E[\frac{\partial L}{\partial \mu_i} \times \frac{\partial L}{\partial \mu_j}] \\ &= \frac{1}{4}E[\{[\Sigma^{-1} + (\Sigma^{-1})^T](y-\mu)\}_i \times \{[\Sigma^{-1} + (\Sigma^{-1})^T](y-\mu)\}_j] \\ &= E[\{\Sigma^{-1}(y-\mu)\}_i \times \{\Sigma^{-1}(y-\mu)\}_j] \\ &= E[[\{\Sigma^{-1}(y-\mu)\}\{\Sigma^{-1}(y-\mu)\}^T]_{ij}] \\ &= E[\Sigma^{-1}(y-\mu)\}(y-\mu)^T\Sigma^{-1}]_{ij} \\ &= [\Sigma^{-1}E[(y-\mu)\}(y-\mu)^T]\Sigma^{-1}]_{ij} \\ &= [\Sigma^{-1}\Sigma\Sigma^{-1}]_{ij} \\ &= \Sigma^{-1}_{ij} \end{aligned} \tag{B.38}$$

Case 2. $i > d,\ j < d$ For brevity of notation, let $p_{(i-d)} = \tilde{p}_i,\ q_{(i-d)} = \tilde{q}_i$. Thus,

$$\begin{aligned} G_{ij} &= E[\frac{\partial L}{\partial \Sigma_{\tilde{p}_i,\tilde{q}_i}} \times \frac{\partial L}{\partial \mu_j}] \\ &= \frac{1}{2}E[\{-\Sigma^{-1} + [\Sigma^{-1}(y-\mu)(y-\mu)^T\Sigma^{-1}]\}_{\tilde{p}_i,\tilde{q}_i}\{\Sigma^{-1}(y-\mu)\}_j] \\ &= \frac{1}{2}E[\{-\Sigma^{-1}]\}_{\tilde{p}_i,\tilde{q}_i}\{\Sigma^{-1}(y-\mu)\}_j] + \frac{1}{2}E[\{[\Sigma^{-1}(y-\mu)(y-\mu)^T\Sigma^{-1}]\}_{\tilde{p}_i\tilde{q}_i}\{\Sigma^{-1}(y-\mu)\}_j] \\ &= 0 + \frac{1}{2}E[\{[\Sigma^{-1}(y-\mu)(y-\mu)^T\Sigma^{-1}]\}_{\tilde{p}_i\tilde{q}_i}\{\Sigma^{-1}(y-\mu)\}_j] \\ &= \frac{1}{2}E[\Sigma^{-1}_{\tilde{p}_i r}(y-\mu)_r(y-\mu)_s\Sigma^{-1}_{s\tilde{q}_i}\Sigma^{-1}_{jk}(y-\mu)_k] \\ &= \frac{1}{2}\Sigma^{-1}_{\tilde{p}_i r}\Sigma^{-1}_{s\tilde{q}_i}\Sigma^{-1}_{jk}E[(y-\mu)_r(y-\mu)_s(y-\mu)_k] \end{aligned} \tag{B.39}$$

Case 3. $i < d,\ j > d$

Again, let $p_{j-d} = \tilde{p}_j$, $q_{j-d} = \tilde{q}_j$

$$\begin{aligned} G_{ij} &= E[\frac{\partial L}{\partial \mu_i} \times \frac{\partial L}{\partial \Sigma_{\tilde{p}_j \tilde{q}_j}}] \\ &= \frac{1}{2} E[\{\Sigma^{-1}(y-\mu)\}_i \{-\Sigma^{-1} + [\Sigma^{-1}(y-\mu)(y-\mu)^T \Sigma^{-1}]\}_{\tilde{p}_j \tilde{q}_j}] \\ &= \frac{1}{2} E[\{-\Sigma^{-1}]\}_{\tilde{p}_j \tilde{q}_j} \{\Sigma^{-1}(y-\mu)\}_i] + \frac{1}{2} E[\{[\Sigma^{-1}(y-\mu)(y-\mu)^T \Sigma^{-1}]\}_{\tilde{p}_j \tilde{q}_j} \{\Sigma^{-1}(y-\mu)\}_i] \\ &= 0 + \frac{1}{2} E[\{[\Sigma^{-1}(y-\mu)(y-\mu)^T \Sigma^{-1}]\}_{\tilde{p}_j \tilde{q}_j} \{\Sigma^{-1}(y-\mu)\}_i] \\ &= 0 \end{aligned} \tag{B.40}$$

Case 4. $i, j > d$

$$\begin{aligned} G_{ij} &= E[\frac{\partial L}{\partial \Sigma_{\tilde{p}_i \tilde{q}_i}} \times \frac{\partial L}{\partial \Sigma_{\tilde{p}_j \tilde{q}_j}}] \\ &= \frac{1}{4} E[\{-\Sigma^{-1} + [\Sigma^{-1}(y-\mu)(y-\mu)^T \Sigma^{-1}]\}_{\tilde{p}_i \tilde{q}_i} \{-\Sigma^{-1} + [\Sigma^{-1}(y-\mu)(y-\mu)^T \Sigma^{-1}]\}_{\tilde{p}_j \tilde{q}_j}] \\ &= \frac{1}{4} E[\Sigma^{-1}_{\tilde{p}_i \tilde{q}_i} \Sigma^{-1}_{\tilde{p}_j \tilde{q}_j}] - \frac{1}{4} E[\Sigma^{-1}_{\tilde{p}_i, \tilde{q}_i} [\Sigma^{-1}(y-\mu)(y-\mu)^T \Sigma^{-1}]_{\tilde{p}_j, \tilde{q}_j}] \\ &\quad - \frac{1}{4} E[\Sigma^{-1}(y-\mu)(y-\mu)^T \Sigma^{-1}]_{\tilde{p}_i \tilde{q}_i} \Sigma^{-1}_{\tilde{p}_j \tilde{q}_j}] \\ &\quad + \frac{1}{4} E\{[\Sigma^{-1}(y-\mu)(y-\mu)^T \Sigma^{-1}]_{\tilde{p}_i \tilde{q}_i} [\Sigma^{-1}(y-\mu)(y-\mu)^T \Sigma^{-1}]_{\tilde{p}_j \tilde{q}_j}\} \\ &= -\frac{1}{4} \Sigma^{-1}_{\tilde{p}_i \tilde{q}_i} \Sigma^{-1}_{\tilde{p}_j \tilde{q}_j} + \frac{1}{4} \Sigma^{-1}_{\tilde{p}_i m} \Sigma^{-1}_{n \tilde{q}_i} \Sigma^{-1}_{\tilde{p}_j r} \Sigma^{-1}_{s \tilde{q}_j} E\{(y-\mu)_m (y-\mu)_n (y-\mu)_r (y-\mu)_s\} \end{aligned} \tag{B.41}$$

The central k-order moments of the variable X are gives as follows; see [162]:

(a) If k is odd, $\mu_{1,\ldots,k}(X-\xi) = 0$

(b) If k is even with $k = 2\lambda$, then it is $\mu_{1,2,\ldots,2\lambda}(X-\xi) = \sum c_{ij} c_{kl} \ldots c_{xz}$, where the sum is taken over all permutations of $\{1, 2, \ldots 2\lambda\}$ giving $(2\lambda-1)!/2^(\lambda-1)(\lambda-1)!$ terms in the sum, each being the product of λ covariance. Therefore, the 4-order moments are given by

$$E(X_i^4) = 3c_{ii} \tag{B.42}$$
$$E(X_i^3 X_j) = 3c_{ii}c_{jj}$$
$$E(X_i^2 X_j^2) = c_{ii}c_{jj} + 2c_{ij}^2$$
$$E(X_i^2 X_j X_p) = c_{ii}c_{jp} + 2c_{ij}c_{ip}$$
$$E(X_i X_j X_p X_q) = c_{ij}c_{pq} + c_{ip}c_{jq} + c_{iq}c_{jp}$$

Now, consider the last term on the RHS of equation B.41,

$$\phi(\tilde{p}_i, \tilde{p}_j, \tilde{q}_i, \tilde{q}_j) = \Sigma^{-1}_{\tilde{p}_i m}\Sigma^{-1}_{n\tilde{p}_j}\Sigma^{-1}_{\tilde{q}_i r}\Sigma^{-1}_{s\tilde{q}_j} E\{(y-\mu)_m(y-\mu)_n(y-\mu)_r(y-\mu)_s\} \tag{B.43}$$

From the last equation in B.42, we have

$$\begin{aligned}
\phi(\tilde{p}_i, \tilde{p}_j, \tilde{q}_i, \tilde{q}_j) &= \Sigma^{-1}_{\tilde{p}_i m}\Sigma^{-1}_{n\tilde{q}_i}\Sigma^{-1}_{\tilde{p}_j r}\Sigma^{-1}_{s\tilde{q}_j}(\Sigma_{mn}\Sigma_{rs} + \Sigma_{mr}\Sigma_{ns} + \Sigma_{ms}\Sigma_{nr}) \qquad \text{(B.44)}\\
&= \Sigma^{-1}_{\tilde{p}_i m}\Sigma^{-1}_{n\tilde{q}_i}\Sigma^{-1}_{\tilde{p}_j r}\Sigma^{-1}_{s\tilde{q}_j}\Sigma_{mn}\Sigma_{rs} + \Sigma^{-1}_{\tilde{p}_i m}\Sigma^{-1}_{n\tilde{q}_i}\Sigma^{-1}_{\tilde{p}_j r}\Sigma^{-1}_{s\tilde{q}_j}\Sigma_{mr}\Sigma_{ns} + \Sigma^{-1}_{\tilde{p}_i m}\Sigma^{-1}_{n\tilde{q}_i}\Sigma^{-1}_{\tilde{p}_j r}\Sigma^{-1}_{s\tilde{q}_j}\Sigma_{ms}\Sigma_{nr}\\
&= \Sigma^{-1}_{\tilde{p}_i m}\Sigma_{mn}\Sigma^{-1}_{n\tilde{q}_i}\Sigma^{-1}_{\tilde{p}_j r}\Sigma_{rs}\Sigma^{-1}_{s\tilde{q}_j} + \Sigma^{-1}_{\tilde{p}_i m}\Sigma_{mr}\Sigma^{-1}_{n\tilde{q}_i}\Sigma^{-1}_{\tilde{p}_j r}\Sigma_{ns}\Sigma^{-1}_{s\tilde{q}_j} + \Sigma^{-1}_{\tilde{p}_i m}\Sigma_{ms}\Sigma^{-1}_{n\tilde{q}_i}\Sigma^{-1}_{\tilde{p}_j r}\Sigma^{-1}_{s\tilde{q}_j}\Sigma_{nr}\\
&= \delta_{\tilde{p}_i n}\Sigma^{-1}_{n\tilde{q}_i}\delta_{\tilde{p}_j s}\Sigma^{-1}_{s\tilde{q}_j} + \delta_{\tilde{p}_i r}\Sigma^{-1}_{\tilde{p}_j r}\Sigma^{-1}_{n\tilde{q}_i}\delta_{n\tilde{q}_j} + \delta_{\tilde{p}_i s}\Sigma^{-1}_{s\tilde{q}_j}\Sigma^{-1}_{n\tilde{q}_i}\delta_{\tilde{p}_j n}\\
&= \Sigma^{-1}_{\tilde{p}_i\tilde{q}_i}\Sigma^{-1}_{\tilde{p}_j\tilde{q}_j} + \Sigma^{-1}_{\tilde{p}_j\tilde{p}_i}\Sigma^{-1}_{\tilde{q}_j\tilde{q}_i} + \Sigma^{-1}_{\tilde{p}_i\tilde{q}_j}\Sigma^{-1}_{\tilde{p}_j\tilde{q}_i}
\end{aligned}$$

Note, the assumption that sigma is symmetric has been used in the above equation. Substituting the result above in equation B.41, we get

$$G_{ij} = \frac{1}{4}[\Sigma^{-1}_{\tilde{p}_j\tilde{p}_i}\Sigma^{-1}_{\tilde{q}_j\tilde{q}_i} + \Sigma^{-1}_{\tilde{p}_i\tilde{q}_j}\Sigma^{-1}_{\tilde{p}_j\tilde{q}_i}] \tag{B.45}$$

Derivative of G:

From equation B.38 we have

$$
\begin{aligned}
G_{ij} &= E[\frac{\partial L}{\partial \mu_i} \times \frac{\partial L}{\partial \mu_j}] \qquad \text{(B.46)} \\
&= \frac{1}{4} E[\{[\Sigma^{-1} + (\Sigma^{-1})^T](y-\mu)\}_i \times \{[\Sigma^{-1} + (\Sigma^{-1})^T](y-\mu)\}_j] \\
&= \frac{1}{4} E[\{[\Sigma^{-1} + (\Sigma^{-1})^T]_{im}(y-\mu)_m\} \times \{[\Sigma^{-1} + (\Sigma^{-1})^T]_{jn}(y-\mu)_n\}] \\
&= \frac{1}{4} E[(y-\mu)_m(y-\mu)_n][\Sigma^{-1} + (\Sigma^{-1})^T]_{im}[\Sigma^{-1} + (\Sigma^{-1})^T]_{jn} \\
&= \frac{1}{4} \Sigma_{mn}[\Sigma^{-1} + (\Sigma^{-1})^T]_{im}[\Sigma^{-1} + (\Sigma^{-1})^T]_{jn} \\
&= \frac{1}{4} [\Sigma^{-1} + (\Sigma^{-1})^T]_{im}\Sigma_{mn}[\Sigma^{-1} + (\Sigma^{-1})^T]^T_{nj} \\
&= \frac{1}{4} [\Sigma^{-1} + (\Sigma^{-1})^T]_{im}\Sigma_{mn}[\Sigma^{-1} + (\Sigma^{-1})^T]_{nj} \\
&= \frac{1}{4} \{[\Sigma^{-1} + (\Sigma^{-1})^T]\Sigma[\Sigma^{-1} + (\Sigma^{-1})^T]\}_{ij} \\
&= \frac{1}{4} [\Sigma^{-1}\Sigma\Sigma^{-1} + \Sigma^{-1}\Sigma(\Sigma^{-1})^T + (\Sigma^{-1})^T\Sigma\Sigma^{-1} + (\Sigma^{-1})^T\Sigma(\Sigma^{-1})^T]_{ij} \\
&= \frac{1}{4} [\Sigma^{-1} + 2(\Sigma^{-1})^T + (\Sigma^{-1})^T\Sigma(\Sigma^{-1})^T]_{ij}
\end{aligned}
$$

Since, we have shown earlier, G is a block diagonal matrix. Let us refer to the upper and lower blocks as G_{upper} and G_{lower} Thus,

$$
G_{upper} = \frac{1}{4} \{\Sigma^{-1} + 2(\Sigma^{-1})^T + (\Sigma^{-1})^T\Sigma(\Sigma^{-1})^T\} \qquad \text{(B.47)}
$$

$$
\begin{aligned}
\frac{\partial (G_{upper})_{ij}}{\partial \Sigma_{kl}} &= \frac{1}{4}\frac{\partial [\Sigma^{-1} + 2(\Sigma^{-1})^T + (\Sigma^{-1})^T \Sigma (\Sigma^{-1})^T]_{ij}}{\partial \Sigma_{kl}} \\
&= \frac{1}{4}\frac{\partial [\Sigma^{-1} + 2(\Sigma^{-1})^T]_{ij}}{\partial \Sigma_{kl}} + \frac{1}{4}\frac{\partial [(\Sigma^{-1})^T \Sigma (\Sigma^{-1})^T]_{ij}}{\partial \Sigma_{kl}} \\
&= -\frac{1}{4}[\Sigma^{-1}_{ik}\Sigma^{-1}_{lj} + 2\Sigma^{-1}_{jk}\Sigma^{-1}_{li}] + \frac{1}{4}\frac{\partial [(\Sigma^{-1})^T_{im}\Sigma_{mn}(\Sigma^{-1})^T_{nj}]}{\partial \Sigma_{kl}} \\
&= -\frac{1}{4}[\Sigma^{-1}_{ik}\Sigma^{-1}_{lj} + 2\Sigma^{-1}_{jk}\Sigma^{-1}_{li}] + \frac{1}{4}\frac{\partial (\Sigma^{-1})^T_{im}}{\partial \Sigma_{kl}}\Sigma_{mn}(\Sigma^{-1})^T_{nj}] \\
&\quad + \frac{1}{4}(\Sigma^{-1})^T_{im}\frac{\partial \Sigma_{mn}}{\partial \Sigma_{kl}}(\Sigma^{-1})^T_{nj} + \frac{1}{4}(\Sigma^{-1})^T_{im}\Sigma_{mn}\frac{\partial (\Sigma^{-1})^T_{nj}}{\partial \Sigma_{kl}} \\
&= -\frac{1}{4}[\Sigma^{-1}_{ik}\Sigma^{-1}_{lj} + 2\Sigma^{-1}_{jk}\Sigma^{-1}_{li}] - \frac{1}{4}\Sigma^{-1}_{mk}\Sigma^{-1}_{li}\Sigma_{mn}(\Sigma^{-1})^T_{nj} \\
&\quad + \frac{1}{4}(\Sigma^{-1})^T_{im}\delta_{mk}\delta_{nl}(\Sigma^{-1})^T_{nj} - \frac{1}{4}(\Sigma^{-1})^T_{im}\Sigma_{mn}\Sigma^{-1}_{jk}\Sigma^{-1}_{ln} \\
&= -\frac{1}{4}[\Sigma^{-1}_{ik}\Sigma^{-1}_{lj} + 2\Sigma^{-1}_{jk}\Sigma^{-1}_{li}] - \frac{1}{4}(\Sigma^{-1})^T_{km}\Sigma_{mn}(\Sigma^{-1})^T_{nj}\Sigma^{-1}_{li} \\
&\quad + \frac{1}{4}(\Sigma^{-1})^T_{ik}(\Sigma^{-1})^T_{lj} - \frac{1}{4}(\Sigma^{-1})^T_{im}\Sigma_{mn}(\Sigma^{-1})^T_{nl}\Sigma^{-1}_{jk} \\
&= -\frac{1}{4}[\Sigma^{-1}_{ik}\Sigma^{-1}_{lj} + 2\Sigma^{-1}_{jk}\Sigma^{-1}_{li}] - \frac{1}{4}[(\Sigma^{-1})^T\Sigma(\Sigma^{-1})^T]_{kj}\Sigma^{-1}_{li} \\
&\quad + \frac{1}{4}(\Sigma^{-1})^T_{ik}(\Sigma^{-1})^T_{lj} - \frac{1}{4}[(\Sigma^{-1})^T\Sigma(\Sigma^{-1})^T]_{il}\Sigma^{-1}_{jk} \\
&= -\frac{1}{2}\Sigma^{-1}_{jk}\Sigma^{-1}_{li} - \frac{1}{4}\Sigma^{-1}_{ik}\Sigma^{-1}_{lj} + \frac{1}{4}(\Sigma^{-1})^T_{ik}(\Sigma^{-1})^T_{lj} \\
&\quad - \frac{1}{4}[(\Sigma^{-1})^T\Sigma(\Sigma^{-1})^T]_{kj}\Sigma^{-1}_{li} - \frac{1}{4}[(\Sigma^{-1})^T\Sigma(\Sigma^{-1})^T]_{il}\Sigma^{-1}_{jk}
\end{aligned}
\tag{B.48}
$$

Now, assuming Σ is symmetric, we have

$$
\frac{\partial (G_{upper})_{ij}}{\partial \Sigma_{kl}} = -\Sigma^{-1}_{jk}\Sigma^{-1}_{li} \tag{B.49}
$$

From equation B.45,and without assuming Σ to be symmetric, we have

$$
G_{ij} = \frac{1}{4}[\Sigma^{-1}_{\tilde{p}_j\tilde{p}_i}\Sigma^{-1}_{\tilde{q}_j\tilde{q}_i} + \Sigma^{-1}_{\tilde{p}_i\tilde{q}_j}\{(\Sigma^{-1})^T\Sigma(\Sigma^{-1})^T\}_{\tilde{q}_i\tilde{p}_j}] \tag{B.50}
$$

Therefore, for (k,l) pairs of indices of lower triangular matrix, we have

$$\begin{aligned}
\frac{\partial (G_{lower})_{ij}}{\partial \Sigma_{kl}} &= \frac{1}{4}\frac{\partial\{[\Sigma^{-1}_{\tilde{p}_j\tilde{p}_i}\Sigma^{-1}_{\tilde{q}_j\tilde{q}_i}+\Sigma^{-1}_{\tilde{p}_i\tilde{q}_j}\{(\Sigma^{-1})^T\Sigma(\Sigma^{-1})^T\}_{\tilde{q}_i\tilde{p}_j}]\}}{\partial\Sigma_{kl}} \\
&= \frac{1}{4}\frac{\partial[\Sigma^{-1}_{\tilde{p}_j\tilde{p}_i}\Sigma^{-1}_{\tilde{q}_j\tilde{q}_i}]}{\partial\Sigma_{kl}}+\frac{1}{4}\frac{\partial[\Sigma^{-1}_{\tilde{p}_i\tilde{q}_j}\{(\Sigma^{-1})^T\Sigma(\Sigma^{-1})^T\}_{\tilde{q}_i\tilde{p}_j}]}{\partial\Sigma_{kl}} \\
&= \frac{1}{4}\frac{\partial\Sigma^{-1}_{\tilde{p}_j\tilde{p}_i}}{\partial\Sigma_{kl}}\Sigma^{-1}_{\tilde{q}_j\tilde{q}_i}+\frac{1}{4}\Sigma^{-1}_{\tilde{p}_j\tilde{p}_i}\frac{\partial\Sigma^{-1}_{\tilde{q}_j\tilde{q}_i}}{\partial\Sigma_{kl}}+\frac{1}{4}\frac{\partial\Sigma^{-1}_{\tilde{p}_i\tilde{q}_j}}{\partial\Sigma_{kl}}\{(\Sigma^{-1})^T\Sigma(\Sigma^{-1})^T\}_{\tilde{q}_i\tilde{p}_j} \\
&\quad +\frac{1}{4}\Sigma^{-1}_{\tilde{p}_i\tilde{q}_j}\frac{\partial\{(\Sigma^{-1})^T\Sigma(\Sigma^{-1})^T\}_{\tilde{q}_i\tilde{p}_j}}{\partial\Sigma_{kl}} \\
&= -\frac{1}{4}\Sigma^{-1}_{\tilde{p}_jk}\Sigma^{-1}_{l\tilde{p}_i}\Sigma^{-1}_{\tilde{q}_j\tilde{q}_i}-\frac{1}{4}\Sigma^{-1}_{\tilde{p}_j\tilde{p}_i}\Sigma^{-1}_{\tilde{q}_jk}\Sigma^{-1}_{l\tilde{q}_i}-\frac{1}{4}\Sigma^{-1}_{\tilde{p}_ik}\Sigma^{-1}_{l\tilde{q}_j}\{(\Sigma^{-1})^T\Sigma(\Sigma^{-1})^T\}_{\tilde{q}_i\tilde{p}_j} \\
&\quad +\frac{1}{4}\Sigma^{-1}_{\tilde{p}_i\tilde{q}_j}\{-[(\Sigma^{-1})^T\Sigma(\Sigma^{-1})^T]_{k\tilde{p}_j}\Sigma^{-1}_{l\tilde{q}_i}+(\Sigma^{-1})^T_{\tilde{q}_ik}(\Sigma^{-1})^T_{l\tilde{p}_j}-[(\Sigma^{-1})^T\Sigma(\Sigma^{-1})^T]_{\tilde{q}_il}\Sigma^{-1}_{\tilde{p}_jk}\} \\
&= -\frac{1}{4}\Sigma^{-1}_{\tilde{p}_jk}\Sigma^{-1}_{l\tilde{p}_i}\Sigma^{-1}_{\tilde{q}_j\tilde{q}_i}-\frac{1}{4}\Sigma^{-1}_{\tilde{p}_j\tilde{p}_i}\Sigma^{-1}_{\tilde{q}_jk}\Sigma^{-1}_{l\tilde{q}_i}-\frac{1}{4}\Sigma^{-1}_{\tilde{p}_ik}\Sigma^{-1}_{l\tilde{q}_j}\Sigma^{-1}_{\tilde{q}_i\tilde{p}_j} \\
&\quad +\frac{1}{4}\Sigma^{-1}_{\tilde{p}_i\tilde{q}_j}\{-\Sigma^{-1}_{k\tilde{p}_j}\Sigma^{-1}_{l\tilde{q}_i}+(\Sigma^{-1})^T_{\tilde{q}_ik}(\Sigma^{-1})^T_{l\tilde{p}_j}-\Sigma^{-1}_{\tilde{q}_il}\Sigma^{-1}_{\tilde{p}_jk}\} \\
&= -\frac{1}{4}\Sigma^{-1}_{\tilde{p}_jk}\Sigma^{-1}_{l\tilde{p}_i}\Sigma^{-1}_{\tilde{q}_j\tilde{q}_i}-\frac{1}{4}\Sigma^{-1}_{\tilde{p}_j\tilde{p}_i}\Sigma^{-1}_{\tilde{q}_jk}\Sigma^{-1}_{l\tilde{q}_i}-\frac{1}{4}\Sigma^{-1}_{\tilde{p}_ik}\Sigma^{-1}_{l\tilde{q}_j}\Sigma^{-1}_{\tilde{q}_i\tilde{p}_j} \\
&\quad -\frac{1}{4}\Sigma^{-1}_{\tilde{p}_i\tilde{q}_j}\Sigma^{-1}_{k\tilde{p}_j}\Sigma^{-1}_{l\tilde{q}_i}+\frac{1}{4}\Sigma^{-1}_{\tilde{p}_i\tilde{q}_j}\Sigma^{-1}_{k\tilde{q}_i}\Sigma^{-1}_{\tilde{p}_jl}-\frac{1}{4}\Sigma^{-1}_{\tilde{p}_i\tilde{q}_j}\Sigma^{-1}_{\tilde{q}_il}\Sigma^{-1}_{\tilde{p}_jk}
\end{aligned} \tag{B.51}$$

Connection:

$$\gamma^r_{pq} = \frac{1}{2}g^{rl}\left(\frac{\partial g_{ql}}{\partial\theta_p}+\frac{\partial g_{pl}}{\partial\theta_q}-\frac{\partial g_{pq}}{\partial\theta_l}\right) \ , \ p,q,r,l\in[1,D]$$

Assuming the dimension of the problem is d, the total number of parameters is as follows:

(a) μ_i for $i\in[1...d]$

(b) Σ_{ij} for (i,j) pairs of lower triangular part of the covariance matrix

Now, since G is a function of Σ alone, and the first d parameters in θ correspond to μ,

$$\frac{\partial G}{\partial\theta_i}=0, i\in[1...d]$$

For $i>d$, we have the following possibilities

1. $q,l\in[1...d]$

$\frac{\partial g_{ql}}{\partial \theta_i}$ = (from the equation for $\frac{\partial G_{upper}}{\partial \Sigma}$)

2. one of q,l $\in [1...d]$ and the other is $> d$
   $\frac{\partial g_{ql}}{\partial \theta_i} = 0$

3. both q and l are $> d$
   $\frac{\partial g_{ql}}{\partial \theta_i}$ = (from the equation for for $\frac{\partial G_{lower}}{\partial \Sigma}$)

B.7 Logistic regression problem

$ln(\frac{p}{1-p}) = \beta_0 + \sum_{i=1}^{D} \beta_i X_i$ where p is the success probability. Therefore
Probability $(t = 1) = p(X) = \frac{1}{1+\exp[-(\beta_0+(\beta^T X))]}$
For a data set X consisting of N points and the corresponding values of t, the likelihood of the given vector t for some β is
$p(t|X,\beta) = \prod_{i=1}^{N} (\frac{1}{1+\exp[-(\beta_0+\beta^T X^{(i)})]})^{t_i} \times (1 - \frac{1}{1+\exp[-(\beta_0+\beta^T X^{(i)})]})^{(1-t_i)}$
Log-likelihood: $L = \log(p(t|X,\beta)) = \sum_{i=1}^{N} [t_i \log(\frac{1}{1+\exp[-(\beta_0+\beta^T X^{(i)})]}) + (1 - t_i)\log(1 - \frac{1}{1+\exp[-(\beta_0+\beta^T X^{(i)})]})]$
Append an extra element $(= 1)$ at the beginning of every vector $X^{(i)}$ for convenience, call it $\bar{X}$ such that $\bar{X}^{(i)}$ is now $D+1$ dimensional, hence we have

$$L = \sum_{i=1}^{N} [t_i \log(\frac{1}{1+\exp[-\beta^T \bar{X}^{(i)}]}) + (1-t_i)\log(1 - \frac{1}{1+\exp[-\beta^T \bar{X}^{(i)}]})] \quad \text{(B.52)}$$

Gradient of log-likelihood:

$$\begin{aligned}
\frac{\partial L}{\partial \beta_p} &= \sum_{i=1}^{N}[t_i\frac{\partial(\log(\frac{1}{1+\exp[-\beta^T\bar{X}^{(i)}]}))}{\partial\beta_p}+(1-t_i)\frac{\partial(\log(1-\frac{1}{1+\exp[-\beta^T\bar{X}^{(i)}]})])}{\partial\beta_p}] \quad \text{(B.53)}\\
&= \sum_{i=1}^{N}[-t_i\frac{\partial(\log(1+\exp[-\beta^T\bar{X}^{(i)}]))}{\partial\beta_p}+(1-t_i)\frac{\partial(\log(\exp[-\beta^T\bar{X}^{(i)})])}{\partial\beta_p}\\
&\quad -(1-t_i)\frac{\partial(\log(1+\exp[-\beta^T\bar{X}^{(i)})])}{\partial\beta_p}]\\
&= \sum_{i=1}^{N}[-t_i\frac{\partial(\log(1+\exp[-\beta^T\bar{X}^{(i)}]))}{\partial\beta_p}+(1-t_i)\frac{\partial([-\beta^T\bar{X}^{(i)})])}{\partial\beta_p}\\
&\quad -(1-t_i)\frac{\partial(\log(1+\exp[-\beta^T\bar{X}^{(i)})])}{\partial\beta_p}]\\
&= \sum_{i=1}^{N}[-\frac{\partial(\log(1+\exp[-\beta^T\bar{X}^{(i)}]))}{\partial\beta_p}-(1-t_i)\frac{\partial(\beta^T\bar{X}^{(i)})}{\partial\beta_p}]\\
&= \sum_{i=1}^{N}[\left(-\frac{1}{1+\exp(-\beta^T\bar{X}^{(i)})}\right)\frac{\partial(1+\exp(-\beta^T\bar{X}^{(i)}))}{\partial\beta_p}-(1-t_i)\frac{\partial(\beta^T\bar{X}^{(i)})}{\partial\beta_p}]
\end{aligned}$$

Now,

$$\begin{aligned}
\frac{\partial\{1+\exp(-\beta^T\bar{X}^{(j)})\}}{\partial\beta_p} &= \frac{\partial[\exp(-\beta^T\bar{X}^{(j)})]}{\partial\beta_p}\\
&= \frac{\partial[\exp(-\beta_i\bar{X}_i^{(j)})]}{\partial\beta_p}\\
&= [\exp(-\beta_i\bar{X}_i^{(j)})]\frac{\partial(-\beta_i\bar{X}_i^{(j)})}{\partial\beta_p}\\
&= -[\exp(-\beta_i\bar{X}_i^{(j)})]\bar{X}_p^{(j)}\\
&= -\exp(-\beta^T\bar{X}^{(j)})\bar{X}_p^{(j)}
\end{aligned}$$

Substituting in B.53, we get

$$\frac{\partial L}{\partial \beta_p} = \sum_{i=1}^{N}[\frac{\exp(-\beta^T\bar{X}^{(i)})\bar{X}_p^{(i)}}{1+\exp(-\beta^T\bar{X}^{(i)})}-(1-t_i)\bar{X}_p^{(i)}] \quad \text{(B.54)}$$

Fisher-Information matrix:

For a given $\{X^i, t\}$ pair, we have

$$\begin{aligned} G &= E\left(\frac{\partial L}{\partial \beta_p}\frac{\partial L}{\partial \beta_q}\right) \qquad \text{(B.55)} \\ &= E\left([\frac{\exp(-\beta^T \bar{X}^{(i)})\bar{X}_p^{(i)}}{1+\exp(-\beta^T \bar{X}^{(i)})} - (1-t_i)\bar{X}_p^{(i)}][\frac{\exp(-\beta^T \bar{X}^{(i)})\bar{X}_q^{(i)}}{1+\exp(-\beta^T \bar{X}^{(i)})} - (1-t_i)\bar{X}_q^{(i)}]\right) \\ &= \frac{(\exp(-\beta^T \bar{X}^{(i)}))^2 \bar{X}_p^{(i)}\bar{X}_q^{(i)}}{(1+\exp(-\beta^T \bar{X}^{(i)}))^2} - \frac{2\exp(-\beta^T \bar{X}^{(i)})\bar{X}_p^{(i)}\bar{X}_q^{(i)}}{(1+\exp(-\beta^T \bar{X}^{(i)}))}E(1-t) \\ &\quad +\bar{X}_p^{(i)}\bar{X}_q^{(i)}E[1-2t+t^2] \end{aligned}$$

t is a Bernoulli random variable, therefore

$$\begin{aligned} E(t) &= 1\times p(t=1)+0\times p(t=0) \\ &= p(t=1) \\ &= \frac{1}{1+\exp(-\beta^T \bar{X}^i)} \\ E(t^2) &= 1^2\times p(t=1)+0\times p(t=0) \\ &= p(t=1) \\ &= \frac{1}{1+\exp(-\beta^T \bar{X}^i)} \end{aligned}$$

Substituting in equation (B.55), we have

$$\begin{aligned} G(barX^{(i)},\beta) &= \frac{(\exp(-\beta^T \bar{X}^{(i)}))^2 \bar{X}_p^{(i)}\bar{X}_q^{(i)}}{(1+\exp(-\beta^T \bar{X}^{(i)}))^2} - \frac{2\exp(-\beta^T \bar{X}^{(i)})\bar{X}_p^{(i)}\bar{X}_q^{(i)}}{(1+\exp(-\beta^T \bar{X}^{(i)}))}\left(1-\frac{1}{1+\exp(-\beta^T \bar{X}^i)}\right) \\ &\quad +\bar{X}_p^{(i)}\bar{X}_q^{(i)}\left(1-\frac{2}{1+\exp(-\beta^T \bar{X}^i)}+\frac{1}{1+\exp(-\beta^T \bar{X}^i)}\right) \\ &= \frac{(\exp(-\beta^T \bar{X}^{(i)}))^2 \bar{X}_p^{(i)}\bar{X}_q^{(i)}}{(1+\exp(-\beta^T \bar{X}^{(i)}))^2} - \frac{2(\exp(-\beta^T \bar{X}^{(i)}))^2\bar{X}_p^{(i)}\bar{X}_q^{(i)}}{(1+\exp(-\beta^T \bar{X}^{(i)}))^2} \\ &\quad +\bar{X}_p^{(i)}\bar{X}_q^{(i)}\frac{\exp(-\beta^T \bar{X}^i)}{1+\exp(-\beta^T \bar{X}^i)} \\ &= -\frac{(\exp(-\beta^T \bar{X}^{(i)}))^2 \bar{X}_p^{(i)}\bar{X}_q^{(i)}}{(1+\exp(-\beta^T \bar{X}^{(i)}))^2} + \frac{\exp(-\beta^T \bar{X}^i)(1+\exp(-\beta^T \bar{X}^i))\bar{X}_p^{(i)}\bar{X}_q^{(i)}}{(1+\exp(-\beta^T \bar{X}^i))^2} \\ &= \frac{\exp(-\beta^T \bar{X}^i)\bar{X}_p^{(i)}\bar{X}_q^{(i)}}{(1+\exp(-\beta^T \bar{X}^i))^2} \end{aligned}$$

Therefore, G for the product of likelihoods is, $G(\bar{X}) = \sum_{i=1}^{N} G(\bar{X}^{(i)}, \beta)$

Derivative of Fisher-Information matrix:

$$\frac{\partial G(\bar{X},\beta)_{pq}}{\partial \beta_r} = \sum_{i=1}^{N} \frac{\partial}{\partial \beta_r}\left(\frac{\exp(-\beta^T \bar{X}^{(i)})\bar{X}_p^{(i)}\bar{X}_q^{(i)}}{(1+\exp(-\beta^T \bar{X}^i))^2}\right) \tag{B.56}$$

$$= \sum_{i=1}^{N} \frac{\partial \exp(-\beta^T \bar{X}^{(i)})}{\partial \beta_r}\left(\frac{\bar{X}_p^{(i)}\bar{X}_q^{(i)}}{(1+\exp(-\beta^T \bar{X}^i))^2}\right) + \sum_{i=1}^{N} \frac{\partial}{\partial \beta_r}\left(\frac{1}{(1+\exp(-\beta^T \bar{X}^i))^2}\right)\exp(-\beta^T \bar{X}^i)\bar{X}_p^{(i)}\bar{X}_q^{(i)} \tag{B.57}$$

Now

$$\frac{\partial[\exp(-\beta^T \bar{X}^{(i)})]}{\partial \beta_r} = -\exp(-\beta^T \bar{X}^{(i)})X_r^{(i)}$$

$$\text{and } \frac{\partial}{\partial \beta_r}\frac{1}{\{1+\exp(-\beta^T X^{(j)})\}^2} = -\frac{2}{\{1+\exp(-\beta^T \bar{X}^{(i)})\}^3}\frac{\partial\{1+\exp(-\beta^T \bar{X}^{(i)})\}}{\partial \beta_r}$$

$$= -\frac{2}{\{1+\exp(-\beta^T \bar{X}^{(i)})\}^3}(-\exp(-\beta^T \bar{X}^{(i)})\bar{X}_r^{(i)})$$

$$= 2\frac{\exp(-\beta^T \bar{X}^{(i)})\bar{X}_r^{(i)}}{\{1+\exp(-\beta^T \bar{X}^{(i)})\}^3} \tag{B.58}$$

Substituting in equation (B.56), we have

$$\frac{\partial G_{pq}}{\partial \beta_r} = -\sum_{i=1}^{N}\exp(-\beta^T \bar{X}^{(i)})X_r^{(i)}\left(\frac{\bar{X}_p^{(i)}\bar{X}_q^{(i)}}{(1+\exp(-\beta^T \bar{X}^i))^2}\right) + \sum_{i=1}^{N} 2\frac{\exp(-\beta^T \bar{X}^{(i)})\bar{X}_r^{(i)}}{\{1+\exp(-\beta^T \bar{X}^{(i)})\}^3}\exp(-\beta^T \bar{X}^i)\bar{X}_p^{(i)}\bar{X}_q^{(i)}$$

$$= -\sum_{i=1}^{N}\frac{\exp(-\beta^T \bar{X}^{(i)})\bar{X}_p^{(i)}\bar{X}_q^{(i)}X_r^{(i)}}{(1+\exp(-\beta^T \bar{X}^i))^2} + \sum_{i=1}^{N} 2\frac{(\exp(-\beta^T \bar{X}^{(i)}))^2\bar{X}_p^{(i)}\bar{X}_q^{(i)}\bar{X}_r^{(i)}}{\{1+\exp(-\beta^T \bar{X}^{(i)})\}^3}$$

Connection: $\gamma_{ij}^k = \frac{1}{2}g^{kl}(\partial_i g_{jl} + \partial_j g_{il} - \partial_l g_{ij})$

Bibliography

[1] Arnak Dalalyan. Further and stronger analogy between sampling and optimization: Langevin Monte Carlo and gradient descent. In Conference on Learning Theory, pages 678–689. PMLR, 2017.

[2] Alain Durmus, Eric Moulines, et al. Nonasymptotic convergence analysis for the unadjusted Langevin algorithm. The Annals of Applied Probability, 27(3):1551–1587, 2017.

[3] David E. Golberg. Genetic algorithms in search, optimization, and machine learning. Addion Wesley, 1989(102):36, 1989.

[4] Gareth O. Roberts, Richard L. Tweedie, et al. Exponential convergence of Langevin distributions and their discrete approximations. Bernoulli, 2(4):341–363, 1996.

[5] Robert F. Stengel. Optimal control and estimation. Courier Corporation, 1994.

[6] F. C. Klebaner. Introduction to stochastic calculus with applications. Imperial College Press, third edition, 2012.

[7] Shūichi Nosé. A molecular dynamics method for simulations in the canonical ensemble. Molecular physics, 52(2):255–268, 1984.

[8] Mariya Mamajiwala and Debasish Roy. Stochastic dynamical systems developed on Riemannian manifolds. Probabilistic Engineering Mechanics, 67:103179, 2022.

[9] Emmanuel J. Candes and Yaniv Plan. Matrix completion with noise. Proceedings of the IEEE, 98(6):925–936, 2010.

[10] Prateek Jain and Purushottam Kar. Non-convex optimization for machine learning. Found. Trends Mach. Learn., 10(3–4):142–336, dec 2017.

[11] Claude Lemaréchal. Cauchy and the gradient method. Doc Math Extra, 251(254):10, 2012.

[12] Eduard Stiefel. Methods of conjugate gradients for solving linear systems. J. Res. Nat. Bur. Standards, 49:409–435, 1952.

[13] Roger Fletcher. Practical methods of optimization. John Wiley & Sons, 2013.

[14] Saeed Ghadimi and Guanghui Lan. Accelerated gradient methods for non-convex nonlinear and stochastic programming. Mathematical Programming, 156(1):59–99, 2016.

[15] Yurii Nesterov. A method for unconstrained convex minimization problem with the rate of convergence o (1/kˆ 2). In Doklady an ussr, volume 269, pages 543–547, 1983.

[16] James Kennedy and Russell Eberhart. Particle swarm optimization. In Proceedings of ICNN'95-international conference on neural networks, volume 4, pages 1942–1948. IEEE, 1995.

[17] Marco Dorigo and Mauro Birattari. Ant colony optimization. encyclopedia of machine learning, 2010.

[18] Rainer Storn and Kenneth Price. Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. Journal of global optimization, 11(4):341–359, 1997.

[19] Nikolaus Hansen. The CMA evolution strategy: a comparing review. Towards a new evolutionary computation, pages 75–102, 2006.

[20] Hans-Georg Beyer and Bernhard Sendhoff. Covariance matrix adaptation revisited–the CMSA evolution strategy–. In International Conference on Parallel Problem Solving from Nature, pages 123–132. Springer, 2008.

[21] Jing Dong and Xin T. Tong. Replica exchange for non-convex optimization. J. Mach. Learn. Res., 22:173–1, 2021.

[22] Steven T. Smith. Optimization techniques on Riemannian manifolds. Fields institute communications, 3(3):113–135, 1994.

[23] Elton P. Hsu. Stochastic analysis on manifolds, volume 38 of Graduate studies in mathematics ; v. 38. American Mathematical Society, Providence, R.I., 2002.

[24] Ralph L. Cohen. The immersion conjecture for differentiable manifolds. Annals of Mathematics, 122(2):237–328, 1985.

[25] David Elworthy. Geometric aspects of diffusions on manifolds. In École d'Été de Probabilités de Saint-Flour XV–XVII, 1985–87, pages 277–425. Springer, 1988.

[26] Debasish Roy and Gorti V. Rao. Stochastic Dynamics, Filtering and Optimization. Cambridge University Press, 2017.

[27] Joseph Pegna and Franz-Erich Wolter. Surface curve design by orthogonal projection of space curves onto free-form surfaces. 1996.

[28] Ke Tang, Xın Yáo, Ponnuthurai Nagaratnam Suganthan, Cara MacNish, Ying-Ping Chen, Chih-Ming Chen, and Zhenyu Yang. Benchmark functions for the CEC'2008 special session and competition on large scale global optimization. Nature inspired computation and applications laboratory, USTC, China, 24:1–18, 2007.

[29] Pierre Del Moral and Spiridon Penev. Stochastic Processes: From Applications to Theory. Chapman and Hall/CRC, 2017.

[30] John Kent. Time-reversible diffusions. Advances in Applied Probability, 10(4):819–835, 1978.

[31] Kai Lai Chung. Lectures from Markov processes to Brownian motion, volume 249. Springer Science & Business Media, 2013.

[32] Gareth O. Roberts and Osnat Stramer. Langevin diffusions and Metropolis-Hastings algorithms. Methodology And Computing In Applied Probability, 4(4):337–357, Dec 2002.

[33] Mark Girolami and Ben Calderhead. Riemann manifold Langevin and Hamiltonian Monte Carlo methods. Journal Of The Royal Statistical Society Series B-Statistical Methodology, 73(2):123–214, 2011.

[34] Saikat Sarkar, Debasish Roy, and Ram Mohan Vasu. A global optimization paradigm based on change of measures. Royal Society open science, 2(7):150123, 2015.

[35] Harold Kushner and George Yin. Stochastic approximation and recursive algorithms and applications, volume 35. Springer Science & Business Media, 2003.

[36] Mariya Mamajiwala, Debasish Roy, and Serge Guillas. Geometrically adapted Langevin dynamics for Markov chain Monte Carlo simulations. arXiv:2201.08072, 2022.

[37] T. Latzko, W. Wiechert, K. Nöh, and U. Jaekel. Markov Chain Monte Carlo methods to analyze the steady-state flux solution space of metabolic network models. ESM 2012 - 2012 European Simulation and Modelling Conference: Modelling and Simulation 2012, pages 84–90, 01 2012.

[38] L. Martino, V. Elvira, D. Luengo, J. Corander, and F. Louzada. Orthogonal parallel MCMC methods for sampling and optimization. Digital Signal Processing, 58(C):64–84, 2016.

[39] Simon L. Cotter, Gareth O. Roberts, Andrew M. Stuart, and David White. MCMC methods for functions: modifying old algorithms to make them faster. Statistical Science, 28(3):424–446, 2013.

[40] Zia Khan, Tucker Balch, and Frank Dellaert. An MCMC-based particle filter for tracking multiple interacting targets. In Tomás Pajdla and Jiří Matas, editors, Computer Vision - ECCV 2004, pages 279–290. Springer Berlin Heidelberg, 2004.

[41] Yongbo Wang, Huapeng Wu, and Heikki Handroos. Markov Chain Monte Carlo (MCMC) methods for parameter estimation of a novel hybrid redundant robot. Fusion engineering and design, 86(9-11):1863–1867, 2011.

[42] Wolfgang Rieping, Michael Habeck, and Michael Nilges. Inferential structure determination. Science, 309(5732):303–306, 2005.

[43] Jody Hey and Rasmus Nielsen. Integration within the Felsenstein equation for improved Markov chain Monte Carlo methods in population genetics. Proceedings of the National Academy of Sciences, 104(8):2785–2790, 2007.

[44] Christophe Andrieu, Nando De Freitas, Arnaud Doucet, and Michael I. Jordan. An introduction to MCMC for machine learning. Machine learning, 50(1-2):5–43, 2003.

[45] Christian P. Robert and Wu Changye. Markov Chain Monte Carlo methods, a survey with some frequent misunderstandings. arXiv:2001.06249, 2020.

[46] Simon Duane, Anthony D. Kennedy, Brian J. Pendleton, and Duncan Roweth. Hybrid Monte Carlo. Physics Letters B, 195(2):216–222, 1987.

[47] Sébastien Bubeck, Ronen Eldan, and Joseph Lehec. Sampling from a log-concave distribution with projected Langevin Monte Carlo. Discrete & Computational Geometry, 59(4):757–783, 2018.

[48] Marcelo Pereyra. Proximal Markov chain Monte Carlo algorithms. Statistics and Computing, 26(4):745–760, 2016.

[49] Xiang Cheng, Niladri S. Chatterji, Peter L. Bartlett, and Michael I. Jordan. Underdamped Langevin MCMC: A non-asymptotic analysis. In Conference on Learning Theory, pages 300–323, 2018.

[50] Alain Durmus, Eric Moulines, and Marcelo Pereyra. Efficient Bayesian computation by proximal Markov chain Monte Carlo: when Langevin meets Moreau. SIAM Journal on Imaging Sciences, 11(1):473–506, 2018.

[51] Gareth O. Roberts and Jeffrey S. Rosenthal. Optimal scaling of discrete approximations to Langevin diffusions. Journal of the Royal Statistical Society: Series B (Statistical Methodology), 60(1):255–268, 1998.

[52] Laird Arnault Breyer, Mauro Piccioni, Sergio Scarlatti, et al. Optimal scaling of MALA for nonlinear regression. The Annals of Applied Probability, 14(3):1479–1505, 2004.

[53] Natesh S. Pillai, Andrew M. Stuart, Alexandre H. Thiéry, et al. Optimal scaling and diffusion limits for the Langevin algorithm in high dimensions. The Annals of Applied Probability, 22(6):2320–2356, 2012.

[54] Xiang Cheng, Niladri S. Chatterji, Yasin Abbasi-Yadkori, Peter L. Bartlett, and Michael I. Jordan. Sharp convergence rates for Langevin dynamics in the nonconvex setting. arXiv preprint arXiv:1805.01648, 2018.

[55] Raaz Dwivedi, Yuansi Chen, Martin J. Wainwright, and Bin Yu. Log-concave sampling: Metropolis-Hastings algorithms are fast. Journal of Machine Learning Research, 20(183):1–42, 2019.

[56] Arnak Dalalyan, Lionel Riou-Durand, and Avetik Karagulyan. Bounding the error of discretized Langevin algorithms for non-strongly log-concave targets. arXiv:1906.08530, 2019.

[57] Andreas Eberle, Arnaud Guillin, Raphael Zimmer, et al. Couplings and quantitative contraction rates for Langevin dynamics. The Annals of Probability, 47(4):1982–2010, 2019.

[58] Stefan Sommer and Anne Marie Svane. Modelling anisotropic covariance using stochastic development and sub-Riemannian frame bundle geometry. Journal of Geometric Mechanics, 9(3):391, 2017.

[59] Shun-ichi Amari. Natural gradient works efficiently in learning. Neural Computation, 10(2):251–276, 1998.

[60] Anand G. Dabak and Don H. Johnson. Relations between Kullback-Leibler distance and Fisher information. 2002.

[61] Wolfgang Karl Härdle and Léopold Simar. Applied multivariate statistical analysis. Springer Nature, 2019.

[62] Simone Fiori. Nonlinear damped oscillators on Riemannian manifolds: Fundamentals. Journal of Systems Science and Complexity, 29(1):22–40, 2016.

[63] John M. Lee. Riemannian manifolds : an introduction to curvature. Graduate texts in mathematics ; 176. Springer, New York ; London, 1997.

[64] Andy W. C. Lau and Tom C. Lubensky. State-dependent diffusion: Thermodynamic consistency and its path integral formulation. Physical Review E, 76(1):011123, 2007.

[65] Tatiana Xifara, Chris Sherlock, Samuel Livingstone, Simon Byrne, and Mark Girolami. Langevin diffusions and the Metropolis-adjusted Langevin algorithm. Statistics and Probability Letters, 91(C):14–19, 2014.

[66] Ovidiu Calin and Der-Chen Chang. Geometric mechanics on Riemannian manifolds: applications to partial differential equations. Springer Science & Business Media, 2006.

[67] Jerrold E. Marsden and Thomas J. R. Hughes. Mathematical foundations of elasticity. Courier Corporation, 1994.

[68] Mark Zlochin and Yoram Baram. Manifold stochastic dynamics for Bayesian learning. Neural computation, 13(11):2549–2572, 2001.

[69] Simone Fiori. Extended Hamiltonian learning on Riemannian manifolds: Numerical aspects. IEEE Transactions on Neural Networks and Learning Systems, 23(1):7–21, 2011.

[70] Yi-An Ma, Tianqi Chen, and Emily Fox. A complete recipe for stochastic gradient MCMC. In Advances in Neural Information Processing Systems, volume 28. Curran Associates, Inc., 2015.

[71] Samuel Livingstone and Mark Girolami. Information-geometric Markov chain Monte Carlo methods using diffusions. Entropy, 16(6):3074–3102, 2014.

[72] John Nash. C1 isometric imbeddings. Annals of mathematics, pages 383–396, 1954.

[73] Victor Solo. On nonlinear state estimation in a Riemannian manifold. In Proceedings of the 48h IEEE Conference on Decision and Control (CDC) held jointly with 2009 28th Chinese Control Conference, pages 8500–8505. IEEE, 2009.

[74] Matthew D. Hoffman and Andrew Gelman. The No-U-Turn sampler: Adaptively setting path lengths in Hamiltonian Monte Carlo. Journal Of Machine Learning Research, 15:1593–1623, 2014.

[75] Bob Carpenter, Andrew Gelman, Matthew D. Hoffman, Daniel Lee, Ben Goodrich, Michael Betancourt, Marcus Brubaker, Jiqiang Guo, Peter Li, and Allen Riddell. Stan: A probabilistic programming language. Journal of statistical software, 76(1):1–32, 2017.

[76] Paul-Christian Bürkner. brms: An R package for Bayesian multilevel models using Stan. Journal of statistical software, 80(1):1–28, 2017.

[77] Chelsea Muth, Zita Oravecz, and Jonah Gabry. User-friendly Bayesian regression modeling: A tutorial with rstanarm and shinystan. Quantitative Methods for Psychology, 14(2):99–119, 2018.

[78] Jesús María Sanz-Serna. Is the NUTS algorithm correct? arXiv:2005.01336, 2020.

[79] Mariya Mamajiwala and Debasish Roy. Time recursive control of stochastic dynamical systems using forward dynamics and applications. International Journal of Mechanical Sciences, 216:106969, 2022.

[80] Donald E. Kirk. Optimal control theory: an introduction. Courier Corporation, 2004.

[81] Yongbo Peng and Jie Li. Stochastic optimal control of structures. Springer, 2019.

[82] David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dharshan Kumaran, Thore Graepel, et al. A general reinforcement learning algorithm that masters chess, shogi, and go through self-play. Science, 362(6419):1140–1144, 2018.

[83] Kenji Doya. Reinforcement learning in continuous time and space. Neural computation, 12(1):219–245, 2000.

[84] S. Wang, F. Gao, and K. L. Teo. Solving Hamilton-Jacobi-Bellman equations by an upwind finite difference method. In Progress in Optimization, pages 255–268. Springer, 2000.

[85] Iain Smears and Endre Suli. Discontinuous Galerkin finite element approximation of Hamilton–Jacobi–Bellman equations with Cordes coefficients. SIAM Journal on Numerical Analysis, 52(2):993–1016, 2014.

[86] Wei Song and Shirley Dyke. Optimal feedback design for nonlinear stochastic systems using the pseudospectral method. International Journal of Non-Linear Mechanics, 55:70–78, 2013.

[87] Shige Peng and Falei Wang. BSDE, path-dependent PDE and nonlinear Feynman-Kac formula. Science China Mathematics, 59(1):19–36, 2016.

[88] Emanuel Todorov. Efficient computation of optimal actions. Proceedings of the national academy of sciences, 106(28):11478–11483, 2009.

[89] Evangelos Theodorou, Jonas Buchli, and Stefan Schaal. A generalized path integral control approach to reinforcement learning. The Journal of Machine Learning Research, 11:3137–3181, 2010.

[90] Hilbert J Kappen. Path integrals and symmetry breaking for optimal control theory. Journal of statistical mechanics: theory and experiment, 2005(11):P11011, 2005.

[91] Nicolas Perkowski. Backward stochastic differential equations: an introduction. Tanári jegyzet, page 8, 2011.

[92] Huyen Pham. Feynman-Kac representation of fully nonlinear PDEs and applications. Acta Mathematica Vietnamica, 40(2):255–269, 2015.

[93] Ioannis Exarchos and Evangelos A. Theodorou. Stochastic optimal control via forward and backward stochastic differential equations and importance sampling. Automatica, 87:159–165, 2018.

[94] Anthony Le Cavil, Nadia Oudjane, and Francesco Russo. Monte-Carlo algorithms for a forward Feynman–Kac-type representation for semilinear nonconservative partial differential equations. Monte Carlo Methods and Applications, 24(1):55–70, 2018.

[95] Feng Bao, Yanzhao Cao, and Xiaoping Han. Forward backward doubly stochastic differential equations and the optimal filtering of diffusion processes. arXiv preprint arXiv:1509.06352, 18(3):635–661, 2015.

[96] Marcus A. Pereira and Ziyi Wang. Learning deep stochastic optimal control policies using forward-backward sdes. In Robotics: science and systems, 2019.

[97] Marco Fuhrman and Gianmario Tessitore. Existence of optimal stochastic controls and global solutions of forward-backward stochastic differential equations. SIAM Journal on Control and Optimization, 43(3):813–830, 2004.

[98] Shige Peng and Zhen Wu. Fully coupled forward-backward stochastic differential equations and applications to optimal control. SIAM Journal on Control and Optimization, 37(3):825–843, 1999.

[99] Shige Peng. Backward stochastic differential equations and applications to optimal control. Applied Mathematics and Optimization, 27(2):125–144, 1993.

[100] Martino Bardi and Italo Capuzzo-Dolcetta. Optimal control and viscosity solutions of Hamilton-Jacobi-Bellman equations. Springer Science & Business Media, 2008.

[101] Tao Yang and Qingjie Cao. Delay-controlled primary and stochastic resonances of the SD oscillator with stiffness nonlinearities. Mechanical Systems and Signal Processing, 103:216–235, 2018.

[102] Jeff T. Scruggs. An optimal stochastic control theory for distributed energy harvesting networks. Journal of Sound and Vibration, 320(4-5):707–725, 2009.

[103] W. Q. Zhu and Y. J. Wu. Optimal bounded control of harmonically and stochastically excited strongly nonlinear oscillators. Probabilistic engineering mechanics, 20(1):1–9, 2005.

[104] Ji-Hun Park and Kyung-Won Min. Bounded nonlinear stochastic control based on the probability distribution for the sdof oscillator. Journal of sound and vibration, 281(1-2):141–153, 2005.

[105] Stefano Lenci and Giuseppe Rega. Optimal control of nonregular dynamics in a Duffing oscillator. Nonlinear Dynamics, 33(1):71–86, 2003.

[106] Yongxin Chen, Tryphon T. Georgiou, and Michele Pavon. Fast cooling for a system of stochastic oscillators. Journal of Mathematical Physics, 56(11):113302, 2015.

[107] Yongxin Chen, Tryphon T. Georgiou, and Michele Pavon. Optimal steering of a linear stochastic system to a final probability distribution, part i. IEEE Transactions on Automatic Control, 61(5):1158–1169, 2016.

[108] Yaguang Zhu, Shuangjie Zhou, Dongxiao Gao, and Qiong Liu. Synchronization of non-linear oscillators for neurobiologically inspired control on a bionic parallel waist of legged robot. Frontiers in neurorobotics, 13:59, 2019.

[109] Louis M. Pecora and Thomas L. Carroll. Synchronization in chaotic systems. Physical review letters, 64(8):821, 1990.

[110] Muthusamy Lakshmanan. Bifurcations, chaos, controlling and synchronization of certain nonlinear oscillators. In Integrability of Nonlinear Systems, pages 206–255. Springer, 1997.

[111] Thomas E. Murphy, Adam B. Cohen, Bhargava Ravoori, Karl R. B. Schmitt, Anurag V. Setty, Francesco Sorrentino, Caitlin R. S. Williams, Edward Ott, and Rajarshi Roy. Complex dynamics and synchronization of delayed-feedback nonlinear oscillators. Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences, 368(1911):343–366, 2010.

[112] Muthusamy Lakshmanan and Kallummal Murali. Chaos in nonlinear oscillators: controlling and synchronization, volume 13. World scientific, 1996.

[113] Anna Najdecka, Vahid Vaziri, and Marian Wiercigroch. Synchronization control of parametric pendulums for wave energy extraction. In Proceedings of the International Conference on Renewable Energies and Power Quality, 2010.

[114] Aline S. De Paula, Marcelo A Savi, Marian Wiercigroch, and Ekaterina Pavlovskaia. Bifurcation control of a parametric pendulum. International Journal of Bifurcation and Chaos, 22(05):1250111, 2012.

[115] Yang Liu, Marian Wiercigroch, James Ing, and Ekaterina Pavlovskaia. Intermittent control of coexisting attractors. Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences, 371(1993):20120428, 2013.

[116] Yongxin Chen, Tryphon T. Georgiou, and Allen Tannenbaum. Stochastic control and nonequilibrium thermodynamics: Fundamental limits. IEEE Transactions on Automatic Control, 65(7):2979–2991, 2019.

[117] Christopher Jarzynski. Equilibrium free-energy differences from nonequilibrium measurements: A master-equation approach. Physical Review E, 56(5):5018, 1997.

[118] Troy Shinbrot, Celso Grebogi, James A. Yorke, and Edward Ott. Using small perturbations to control chaos. nature, 363(6428):411–417, 1993.

[119] K. Pyragas and A. Tamaševičius. Experimental control of chaos by delayed self-controlling feedback. Physics Letters A, 180(1-2):99–102, 1993.

[120] Ramon Van Handel. Stochastic calculus, filtering, and stochastic control. Course notes., URL http://www. princeton. edu/rvan/acm217/ACM217. pdf, 14, 2007.

[121] Bart Van Den Broek, Wim Wiegerinck, and Bert Kappen. Graphical model inference in optimal control of stochastic multi-agent systems. Journal of Artificial Intelligence Research, 32:95–122, 2008.

[122] Raphaël Chetrite and Hugo Touchette. Nonequilibrium Markov processes conditioned on large deviations. In Annales Henri Poincaré, volume 16, pages 2005–2057. Springer, 2015.

[123] Peter J. Thomas and Benjamin Lindner. Phase descriptions of a multidimensional Ornstein-Uhlenbeck process. Physical Review E, 99(6):062221, 2019.

[124] Edward N. Lorenz. Deterministic nonperiodic flow. Journal of the atmospheric sciences, 20(2):130–141, 1963.

[125] M. K. Sifakis and S. J. Elliott. Strategies for the control of chaos in a Duffing–Holmes oscillator. Mechanical Systems and Signal Processing, 14(6):987–1002, 2000.

[126] Debasish Roy. Explorations of the phase-space linearization method for deterministic and stochastic nonlinear dynamical systems. Nonlinear Dynamics, 23(3):225–258, 2000.

[127] Christopher K. Winkle and L. Mark Berliner. A Bayesian tutorial for data assimilation. Physica D: Nonlinear Phenomena, 230(1):1–16, 2007.

[128] M. Sanjeev Arulampalam, Simon Maskell, Neil Gordon, and Tim Clapp. A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. IEEE Transactions on Signal Processing, 50(2):174–188, 2002.

[129] Michael K. Pitt and Neil Shephard. Filtering via simulation: Auxiliary particle filters. Journal of the American Statistical Association, 94(446):590–599, 1999.

[130] Paul Fearnhead. Using random quasi-Monte-Carlo within particle filters, with application to financial time series. Journal of Computational and Graphical Statistics, 14(4):751–769, 2005.

[131] Mathieu Gerber and Nicolas Chopin. Sequential quasi Monte Carlo. Journal of the Royal Statistical Society: Series B (Statistical Methodology), 77(3):509–579, 2015.

[132] Thomas Bengtsson, Peter Bickel, Bo Li, et al. Curse-of-dimensionality revisited: Collapse of the particle filter in very large scale systems. Probability and statistics: Essays in honor of David A. Freedman, 2:316–334, 2008.

[133] Paul Fearnhead and Hans R. Kuensch. Particle filters and data assimilation. Annual Review of Statistics and Its Application, 5(1):421–449, 2018.

[134] Geir Evensen. Sequential data assimilation with a nonlinear quasi-geostrophic model using Monte Carlo methods to forecast error statistics. Journal of Geophysical Research: Oceans, 99(C5):10143–10162, 1994.

[135] P. L. Houtekamer and Herschel L. Mitchell. Data assimilation using an ensemble Kalman filter technique. Monthly Weather Review, 126(3):796–811, 1998.

[136] Craig Bishop, Brian Etherton, and Sharanya Majumdar. Adaptive sampling with the ensemble transform Kalman filter. part i: Theoretical aspects. Monthly Weather Review, 129(3):420, 2001.

[137] Jeffrey Anderson. An ensemble adjustment Kalman filter for data assimilation. Monthly Weather Review, 129(12):2884, 2001.

[138] Jeffrey Whitaker and Thomas Hamill. Ensemble data assimilation without perturbed observations. Monthly Weather Review, 130(7):1913, 2002.

[139] Geir Evensen. The ensemble Kalman filter: theoretical formulation and practical implementation. Ocean Dynamics, 53(4):343–367, 2003.

[140] Milija Zupanski, Ionel M. Navon, and Dusanka Zupanski. The maximum likelihood ensemble filter as a non-differentiable minimization algorithm. Quarterly Journal of the Royal Meteorological Society, 134(633):1039–1050, 2008.

[141] Lars Nerger, Tijana Janjic, Jens Schröter, and Wolfgang Hiller. A unification of ensemble square root Kalman filters. Monthly Weather Review, 140(7):2335–2345, 2012.

[142] Saikat Sarkar, Shubhankar R. Chowdhury, Mamatha Venugopal, Ram Mohan Vasu, and Debasish Roy. A Kushner–Stratonovich Monte Carlo filter

applied to nonlinear dynamical system identification. Physica D: Nonlinear Phenomena, 270(C):46–59, 2014.

[143] Geir Evensen et al. Data assimilation: the ensemble Kalman filter, volume 2. Springer, 2009.

[144] Christophe Andrieu, Arnaud Doucet, and Vladislav B. Tadić. On-line parameter estimation in general state-space models. In Proceedings of the 44th IEEE Conference on Decision and Control, pages 332–337. IEEE, 2005.

[145] Arnaud Doucet and Vladislav B. Tadić. Parameter estimation in general state-space models using particle methods. Annals of the institute of Statistical Mathematics, 55(2):409–422, 2003.

[146] Hamid Moradkhani, Soroosh Sorooshian, Hoshin V. Gupta, and Paul R. Houser. Dual state–parameter estimation of hydrological models using ensemble kalman filter. Advances in water resources, 28(2):135–147, 2005.

[147] R. Kubo. The fluctuation-dissipation theorem. Reports on progress in physics, 29(1):255, 1966.

[148] Takuto Maeda, Kazushige Obara, Masanao Shinohara, Toshihiko Kanazawa, and Kenji Uehira. Successive estimation of a tsunami wavefield without earthquake source data: A data assimilation approach toward real-time tsunami forecasting. Geophysical Research Letters, 42(19):7923–7932, 2015.

[149] Aditya Riadi Gusman, Anne F. Sheehan, Kenji Satake, Mohammad Heidarzadeh, Iyan Eka Mulia, and Takuto Maeda. Tsunami data assimilation of cascadia seafloor pressure gauge records from the 2012 Haida Gwaii earthquake. Geophysical Research Letters, 43(9):4189–4196, 2016.

[150] Mohammad Heidarzadeh, Yuchen Wang, Kenji Satake, and Iyan E. Mulia. Potential deployment of offshore bottom pressure gauges and adoption of

data assimilation for tsunami warning system in the western Mediterranean sea. Geoscience Letters, 6(1):1–12, 2019.

[151] François Le Gland, Valérie Monbet, and Vu-Duc Tran. Large sample asymptotics for the ensemble Kalman filter. PhD thesis, INRIA, 2009.

[152] Jan Mandel, Loren Cobb, and Jonathan D Beezley. On the convergence of the ensemble Kalman filter. Applications of Mathematics, 56(6):533–541, 2011.

[153] Mark D. Butala, Jonghyun Yun, Yuguo Chen, Richard A. Frazin, and Farzad Kamalabadi. Asymptotic convergence of the ensemble Kalman filter. In 2008 15th IEEE International Conference on Image Processing, pages 825–828. IEEE, 2008.

[154] Robert Zwanzig. Nonlinear generalized Langevin equations. Journal of Statistical Physics, 9(3):215–220, November 1973.

[155] Kelvin Shuangjian Zhang, Gabriel Peyré, Mohamed-Jalal Fadili, and Marcelo Pereyra. Wasserstein control of mirror Langevin Monte Carlo. ArXiv, abs/2002.04363, 2020.

[156] Elton P. Hsu. Quasi-invariance of the Wiener measure on the path space over a compact Riemannian manifold. Journal of Functional Analysis, 134(2):417–450, 1995.

[157] Maziar Raissi, Paris Perdikaris, and George E. Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. Journal of Computational physics, 378:686–707, 2019.

[158] Bruch K. Driver. A Cameron-Martin type quasi-invariance theorem for Brownian motion on a compact Riemannian manifold. Journal of functional analysis, 110(2):272–376, 1992.

[159] Ognian Enchev and Daniel W. Stroock. Towards a Riemannian geometry on the path space over a Riemannian manifold. Journal of Functional Analysis, 134(2):392–416, 1995.

[160] Dominic G. B. Edelen. Applied exterior calculus. Courier Corporation, 2005.

[161] Kaare Brandt Petersen, Michael Syskind Pedersen, et al. The matrix cookbook. Technical University of Denmark, 7(15):510, 2008.

[162] Kostas Triantafyllopoulos. Moments and cumulants of the multivariate real and complex Gaussian distributions. Department of Mathematics, University of Bristol, Version, 12, 2002.